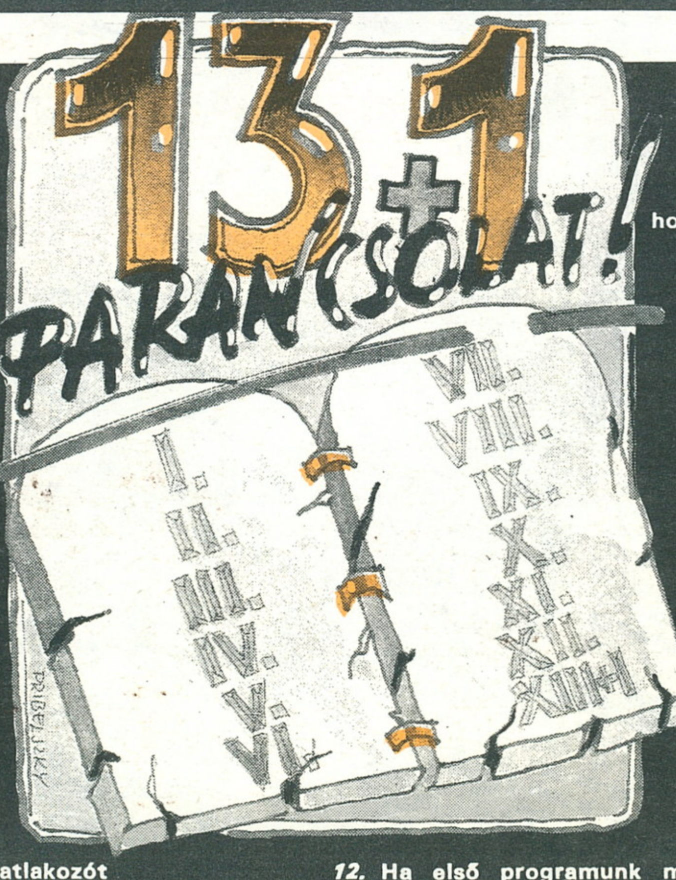
BIT-LET

ötlet SZÁMÍTÁSTECHNIKAI MELLÉKLET · 1987 SZEPTEMBER 48

13+1 PARANCSOLAT!



Ilyenkor nyár végén sok új tagja van a számítógépesek táborának. A túristautakról hazatérők csomagjában nem ritkaság a számítógép. Megszaporodnak ilyenkor a hozzánk érkező levelek, amelyekben kezdők kérnek mindenféle tanácsot. Épp ezért úgy gondoltuk sokan örülnek majd, ha e havi lapszámunkban megpróbáljuk kicsit tréfásan, kicsit komolyan összefoglalni a kezdők 13+1 parancsolatát. Íme:

1. Ne vegyünk olyan gépet, amelyre az égvilágon semmiféle szoftvert sem lehet itthon kapni.

2. Ha már megvettünk egy ilyen gépet:
a) adjunk rajta túl amilyen gyorsan csak lehet
b) hirdessünk az összes lehetséges helyen, hogy cseretársakat keresünk.

3. Mielőtt bekapcsolnánk a gépet olvassuk el a kezelési utasítást, tájékozódjunk a legközelebbi szervíz helyéről és nyitvatartásáról, valamint mérjük fel, hogy milyen módon tudjuk leggyorsabban értesíteni a tűzoltókat.

4. Ha mégsem „szólal" meg a gép, azaz nincs kép a monitoron:
a) azonnal húzzuk ki a hálózati csatlakozót
b) ha a kezelési útmutató újabb áttanulmányozása sem segít, akkor fordítsuk figyelmünket a tévé csatorna beállítójára!

5. Mielőtt egy normál kazettás magnót kapcsolnánk a géphez, győződjünk meg a leírásból, hogy valóban bármilyen magnó csatlakoztatható-e hozzá.

6. Mérjük fel, hogy milyen tárolóegységünk van és min van a program amit szereztünk a géphez. Törődjünk bele, hogy a kazettát a floppy drive-ba, illetve a lemezt a magnóba nem lehet berakni. Ha a művelet mégis sikerülne, vegyük elő a javításhoz a dugipénzünket!

7. Ha mód van rá, akkor elsőként ne olyan programot töltsünk be, amelyhez használati utasításunk is meg nyelvtudásunk is hiányzik, mert ez biztos kudarcélmény.

8. Ha van gyerekünk, akkor az első héten tartsuk távol a géptől, különben mi nem férünk hozzá. Legjobb ha éjjel a gyerek elalvása után vesszük elő a gépet a rejtekhelyéről.

9. A gép első bekapcsolása előtt értesítsük a feleségünket, barátnőnket, esetleg férjünket, barátunkat, hogy néhány hétre elutazunk valami távoli telefon és posta nélküli helyiségbe. (Mondjuk egy ismeretlen afrikai faluba.)

10. A BASIC programozás alapjait ne a géphez adott könyvből akarjuk megtanulni, mert ez általában alkalmatlan erre a célra. Javasoljuk az óvodásoknak szóló könyveket, bár néha még azok is túlságosan magasak.

11. Ha nem értjük a könyvet, amelyből tanulunk, ne magunkban, ne is a számítástechnikában keressük a hibát. Kizárólag a könyvben! Segítségért pedig forduljunk bizalommal bármelyik 30 kiló és 12 év alatti szemüveges fiú ismerősünkhöz.

12. Ha első programunk működik, gondoljunk az ókori csodákra. De azért ne keressük föl programunkkal egyik szoftverházat sem.

13. Ne feledjük: amit mi fölfedezünk, azt nagy valószínűséggel előttünk már néhány millióan fölfedezték.

+1. Ha programjaink írása, futtatása közben váratlan dolgok történnek, amelyek magyarázatát gyakorlott programozó barátaink sem tudják, akkor:
a) már tudjuk, hogy miért hívják gépünket home-computernek
b) olvasgassuk elalvás előtt Murphy törvénykönyvének idevágó fejezetét.

Angyalosi László

BELÜLRŐL

HÍROLDAL



UKLEÁRIS HÁBORÚ

A Massachusetts Institute of Technology, az egyik legjelentősebb amerikai műszaki egyetem szakértői az esetleges nukleáris háború kitörésére vonatkozó különleges tanulmányt készítettek. A százharminchat oldalas anyag négy év alatt, számítógépes szimulációval készült el, amely szerint – feltételezve a reagani űrvédelmi rendszer létezését – a szovjet rakéták mintegy 1%-a jutna el az USA gazdasági célpontjaiig. Viszont már ez a „csekély" mennyiség is olyan károkat idézne elő, aminek helyreállításához kb. huszonöt évre lenne szükség. A kormányszervek vitatják a tanulmány következtetéseinek pontosságát.

HŐERŐMŰ

Félidejéhez érkezett az Oroszlányi Hőerőmű teljesítménynövelő rekonstrukciója. A több mint négymilliárd forint értékű korszerűsítési munkálatokat az elmúlt negyedszázadban teljes kapacitással működő gépek, berendezések elhasználódása tette szükségessé.

A korszerűsítés kiterjed az erőmű irányítástechnikájának fejlesztésére is. A kezelő személyzet munkáját a már felújított egyes és majd valamennyi blokknál mikroszámítógép könnyíti meg, és lehetővé teszi az esetleges üzemzavarok korábbinál jóval gyorsabb és egyszerűbb elhárítását. A rekonstrukciónál nagy figyelmet fordítanak a környezetvédelemre: mind a négy kazánhoz az eddiginél korszerűbb elektronikus pernyeleválasztót építenek be, s az idén tavasszal az erőművet határoló erdősáv körül újabb fákat ültettek.

BRISTOLI BAKI

Igencsak kellemetlen helyzetbe került Bristol város egyik bírája. Haladni akarván a korral, az egyik, általa tárgyalt aktuális ügy részleteit, alakulását, adatait stb. rögtön, közvetlenül számítógépébe vitte be, azzal a céllal, hogy végezetül annak segítségével könnyedén megszerkessze az ügy teljes anyagát. Kezelési tévedésből azonban megsemmisítette a gépben tárolt valamennyi információt. A kellemetlen technikai baki következtében mintegy tízezer font kára keletkezett, és a tárgyalást hosszú időre el kellett napolni.

BÚVÁR

Az Egyesült Államok-beli Battelle Memorial intézet partnert keres, mivel karóra-méretű búvárszámítógépének kereskedelmi változatát is szeretné kifejleszteni. A prototípust készen kapható CMOS mikroprocesszorral és a hozzá tartozó segédáramkörökkel 3,8 cm-es élhosszúságú hibrid áramkörként valósították meg. Számjegyes megjelenítővel és nyomásérzékelővel látták el. A búvárszámítógép jegyzi a víz alatt töltött időt, és szoftvere segíti a búvárt a felszállási sebesség betartásában, a légembólia (keszonbetegség) elkerülésében.

Két éven belül kifejleszthető az 1000 dollárnál olcsóbb kereskedelmi változat.

BONDWELL KÉPTELEFON

Szenzációs, új hang- és képátviteli rendszert, illetve képtelefont ajánl vásárlóinak a hongkongi Bondwell International cég. Az új képtelefon üzemeltethető normál telefonvonalon keresztül is. A készülék működhet videokameraként, digitalizálóként és Hayes kompatíbilis modemként is. Az átvitt képinformáció kinyomtatható, vagy PC kompatíbilis számítógépek memóriájában tárolható.

BERNOULLI

A XVIII. században élt svájci matematikus, Bernoulli törvénye alapján – ami az áramló folyadékok és gázok sebessége és nyomása közötti összefüggéseket mondja ki – az IBM cég három szakembere író/olvasó fejet rögzített egy fémlemezbe, és efölött forog nagy sebességgel az adatlemez. Az így létrejövő szívóhatás lecsökkenti a lemezek közötti légrést és a fej le tudja olvasni, vagy fel tudja írni az adatokat anélkül, hogy mechanikailag érintkezne az adatlemezzel.

FÉNYCHIPEK

Az Egyesült Államokban mintegy tíz éve folyó kutatások eredményeként tesztelhető állapotba kerültek az első fényáramkörök. Az optikai chipek kifejlesztése hamarosan forradalmasítani fogja a nagyteljesítményű számítógépek gyártását. A kutatók véleménye szerint az új optikai alkatelemekkel folytatott kísérletek néhány év múlva elvezetnek egy könnyen gyártható és alkalmazható prototípushoz. A közel százmillió dolláros programban ekkor kezdődhet meg a mai szuperszámítógépnél akár három nagyságrenddel nagyobb kapacitású, párhuzamos elven működő fényszámítógépek megépítése.

DETEKTÍVPROGRAM

Új, nagy tudású társa van az amerikai nyomozóknak, a Scorecard nevet viselő nyomozóprogram. A Ron Wutrich, 28 éves számítógép-programozó által készített szoftver sikeresen vizsgázott. Segítségével kétszáztíz, régóta körözött veszélyes bűnözőt sikerült elfogni, akik közül százhatvanhat kábítószerkereskedő volt. Az alig két hónap alatt elkészült detektívrendszer tulajdonképpen egy relációs adatbázis. Felkutatja a nyomokat és megkeresi a köztük lévő összefüggéseket. Jelzi a szökésben lévő bűnöző feltételezett tartózkodási helyét, megnevezi a legvalószínűbb személyt, aki elvezethet hozzá. A rendszer hasznosítására Washingtonban külön számítóközpont épül. Wutrich pedig dolgozik programja továbbfejlesztésén.

NB I

Történelmi sorsolásra került sor a közelmúltban a Magyar Labdarúgó Szövetség tanácstermében. Ugyanis első ízben készült számítógéppel a labdarúgó NB I teljes menetrendje az 1987. augusztus 15-i első fordulótól az 1988. június 8-i idényzárásig. A számítógépes sorsolást az NB I-es klubok képviselőinek jelenlétében Czékus Lajos, az MLSZ főtitkár-helyettese, Csiki Károly, az MLSZ munkatársa és Szániel János, a liga elnöke vezette le.

-CLN

MOTOR-DIAGNOSZTIKA

Korszerű, számítógéppel összekapcsolt motordiagnosztikai berendezéseket szerzett be a Zala Volán. Az 1500 gépjárművel – köztük 340 autóbusszal – működő szállítási vállalatnál e berendezések birtokában alaposabb műszeres vizsgálat alá vetik a járműveket, kiküszöbölik a motorok túlfogyasztását, s egyszersmind óvják a környezetet attól, hogy túlzott mennyiségű kipufogó gáz kerülhessen a levegőbe. A kezdeti eredmények összegezéséből kitűnik, hogy a számítógépes módszer alkalmazása óta a korábbinál nagyobb fuvarteljesítmény mellett 2,6 százalékkal csökkent az üzemanyagfelhasználás. A gépjárművek vezetői is érdekeltek abban, hogy a rájuk bízott jármű fogyasztása optimális legyen, mert részesülnek a megtakarított üzemanyag értékéből.

KÖNNYŰIPAR

Jelentős eredményeket értek el a számítástechnika alkalmazásával a könnyűipari vállalatoknál. Különösen a gyártáselőkészítési fázisban alkalmazzák a számítógépeket, mert segítségükkel a mintatervezés, a modellrajzok és az úgynevezett terítékrajzok elkészítése jelentősen felgyorsul. A korábbi két-három hetes előkészítési idő mindössze egy-két napra csökken le. Gondot okoz viszont, hogy a könnyűipari vállalatok között nem kellő mértékű az együttműködés a számítástechnikai módszerek alkalmazásában, a sikeres programok cseréjében. E probléma kiküszöbölésére az Ipari Minisztérium koordinálásával még ez évben társulást hoznak létre.



SICONTACT SZOFTVER

A budapesti székhelyű Sicontact magyar–NSZK vegyesvállalat tovább bővíti tevékenységét, bekapcsolódik a hazai, illetve a külföldi értékesítésre szánt szoftverek készítésébe. A vegyesvállalat a számítógépes programok készítésére szoftverházat szervezett. Az új egység a külföldi tulajdonos, az NSZK-beli Siemens megrendelésére is készít speciális számítógépes programokat. A tervek szerint már 1987-ben mintegy húszmillió forint értékű szoftvert exportálnak az NSZK-ba, s emellett lehetővé válik, hogy a Siemens számítógépek hazai felhasználói forintért vásárolhassanak gépeikhez programokat.

BIOPROGRAM

Az ország egész területére kiterjedő komplex számítógépes növényvédelmi szolgálatot hozott létre a pécsi Pannónia Agrárinnovációs Közös Vállalat. A nemzetközi viszonylatban is korszerű eljárás lényege az, hogy a kijelölt táblákon gyomfelvételt készítenek, továbbá begyűjtik a táblára vonatkozó egyéb – talajvizsgálati, agrotechnikai, ökológiai, termelési stb. – adatokat, s ezek figyelembevételével választja ki a számítógép a leghatásosabb és legolcsóbb védekezési technológiát. A gyomkártevők után kidolgozzák a gombák és a rovarok elleni védekezés számítógépes módszerét is. A biológiai programot eddig száz mezőgazdasági üzem vásárolta meg a Pannóniától.

ÚJ!

Omega MC 68020

Az angol Windrush Micro System cég újdonsága az Omega MC 68020 típusú gép, amely Unix típusú 0S9/68 K operációs rendszerével, alapkiépítésben egy négyfelhasználós rendszer. Alapja egy MC 68020 32 bites mikroprocesszor. Sebessége 12,5 MHz. 1 Mbyte-os RAM-ot és 128 Kbyte-os ROM-ot tartalmaz. 25,5 Mbyte-os, 3 1/2 inches Winchester és 1,2 Mbyte-os, szintén 3 1/2 inches floppy meghajtó csatlakozik hozzá.

TÁVVEZÉRELT

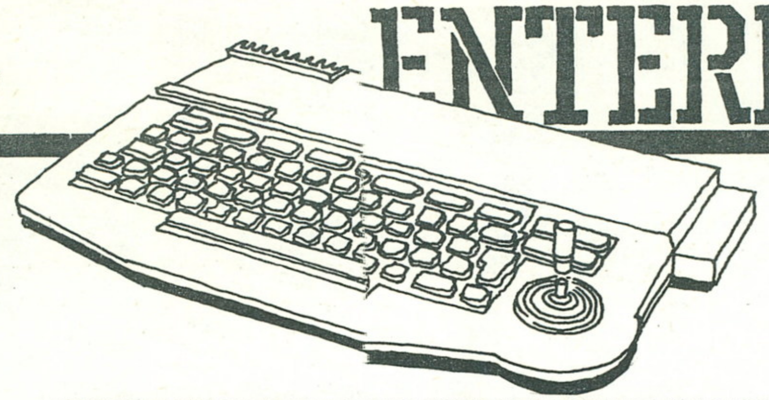
Távvezérelhető helikopter kifejlesztésén dolgoznak a Yamaha cég mérnökei. A gép vezérlését két elektronikus rendszer látja el. Az egyik egy mikroszámítógép, amely a Földről érkező utasítások alapján irányítja a mozgást, a másik egy úgynevezett gyrorendszer, amely a váratlan széllökések kiegyenlítését végzi. A távvezérelt repülő szerkezet sokféle feladatot végezhet: vetőmagok, gyom- és rovarirtók kiszórását, térképészeti felvételek készítését, mentést, esetleg teherszállítást fogókarmok segítségével.

TISZA VOLAN

A Szegedi Tisza Volán Vállalatnál számítógépek, valamint ezekkel összekapcsolt huszonöt képernyős terminál segíti a négy és fél ezernyi dolgozó munkáját, a másfél ezernyi gépjármű éves fuvarbeosztását. Az autóbuszjáratok optimális megtervezését, a menetrendkönyvek felkészítését is a számítástechnika alkalmazásával oldották meg. Computeres menetrend-szerkesztői rendszerüket, valamint forgalomszervezési rendszerüket más társvállalatok is átvették. A gépjárműállomány műszaki állapotának megóvására mikroszámítógépes diagnosztikai műszercsaládot fejlesztettek ki.

Mészáros Gyula:

Mi micsoda az ENTERPRISE® on?



Gondolom, sok Enterprise-tulajdonos várja epekedve a géphez mellékelt kézikönyvben beharangozott Enterprise Technikai Ismertetőt, ebben találhatók ugyanis többek között a címben említett csatlakozó-kiosztások is. A kézikönyv ugyanis semmiféle ilyen információt – a csatlakozók megnevezésén kívül – nem közöl. Mi mindenesetre kibogarásztunk néhány dolgot.

Úgy vélem – gondolva a kis hazánkban föllelhető sokféle típusú gépre és gyakran tanácstalan tulajdonosaikra – a tények, adatok mellett érdekes lehet azok kiderítésének módszerét is ismertetni. Tegyük fel tehát, hogy adott egy számítógép, amelynek különféle csatlakozói vannak. Ismerjük a csatlakozók nevét, tehát hogy körülbelül mire valók, és szeretnénk megtudni, hogyan kell bekötni azokat. A munkához szükségünk lesz egy ellenállásmérőre és egy kb. 15 Volt méréshatárú egyenfeszültségű voltmérőre. Ha sikerül oszcilloszkópot szerezni, az sem baj. Egyetlen dologra érdemes a gép élete érdekében vigyáznunk: a rendszer bővítő csatlakozó(k)hoz (System Bus, Expansion Port, ROM Bay stb.) NE nyúljunk!

1. lépés: Keressük meg a földpontot!

Nagyon fontos! Mindenekelőtt kapcsoljuk ki a gépet, és húzzuk ki belőle a hálózati tápegységet (ha külön van)! Rendszerint minden gépen van többé-kevésbé szabványos koaxiális tv-csatlakozó. Ennek külső, nagyobb fémharangja biztosan földpont, és rendszerint jól hozzáférhető. Az ellenállásmérő negatív pólusát kössük ehhez a ponthoz, a másik pólusával pedig lépegessünk végig a csatlakozók kivezetésein! Kikapcsolt állapotban ez egyik gépnek sem árthat. Az 1 Ohm alatti ellenállású pontok biztosan földpontok. Ha 1 MOhmos vagy nagyobb méréshatárban sem mutat semmit a műszerünk, akkor váltsunk polaritást! Ha így is ugyanaz az eredmény, nagy valószínűséggel a vizsgált láb sehová sem vezet, a továbbiakban nem kell vele foglalkoznunk.

2. lépés: Válasszuk szét a kimeneteket és a bemeneteket!

Ez digitális vonalak esetén egyszerű. Először is kapcsoljuk be a gépet! Egy kb. 10 kOhm-os ellenállással aszerint, hogy az adott kivezetés voltmérővel megmérve alaphelyzetben magas vagy alacsony feszültségszintet mutatott, kössük a kivezetést az ellenkező potenciálhoz! (Tehát, ha például voltmérő egy lábon 2 Voltnál nagyobb feszültséget jelzett terheletlenül, kössük azt a lábat a földhöz.) Ha a gépből nem jön ki a +5 Volt, egy laposelem negatív sarkát a földre kötve a pozitív sarok megfelel a célnak. Három eset lehetséges:

a) az ellenállás csak kicsit változtatja meg a feszültséget, ekkor a mért vonal kimenet,
b) a feszültség ellenkezőjére változik (pl. 3 Voltból 0,1 Volt lesz), ekkor a vonal bemenet,
c) az eredetileg +/–5 vagy +/–12 voltos vonal feszültsége semmit sem csökken, ekkor tápfeszültség-kivezetést találtunk.

Analóg vonalnál a fenti eljárás nem mindig jó, bár bajt itt sem okozhat. De általában a gépeken a magnóbemeneten kívül nem szokott analóg bemenet lenni, így első közelítésként föltehetjük, hogy minden analóg vonal kimenet.

3. lépés: Határozzuk meg a kivezetések funkcióit!

Erre már nemigen lehet általános receptet adni, a követendő eljárás nagyban függ a csatlakozó céljától és az előzőekben megállapítottaktól. Nézzünk néhány esetet:

CENTRONICS csatlakozó: adjuk ki sorban a következő utasításokat:

```
LPRINT CHR$(0);
LPRINT CHR$(1);
LPRINT CHR$(2);
LPRINT CHR$(4);
LPRINT CHR$(8);
LPRINT CHR$(16);
LPRINT CHR$(32);
LPRINT CHR$(64);
LPRINT CHR$(128);
```

és mindegyik utasítás után mérjük meg a kimenetek feszültségeit! Amelyik vonal az első utasítás után magas szinten marad, az a printer STROBE bemenetére kötendő. A további utasításokkal sorban kideríthetjük az adatbitek sorrendjét. Ha csak egy bemenet van, az minden bizonnyal a printer BUSY kimenetére kötendő. Több bemenet esetén próbáljuk azokat egyesével magas szintre kötni míg a többi bemenet alacsony szinten van. Amelyik láb földelésére megszakad a kiírás (lefagy a gép), az lesz a BUSY bemenet.

RS–232 vonal: itt rendszerint két bemenetet és két kimenetet találunk. Próbáljunk a soros vonalra adatot kiküldeni: a gép valószínűleg lefagy. Adjunk magas szintet az egyik bemenetre. Ha a gép elindul, megtaláltuk a CTS (Clear To Send) lábat, ha nem indul, próbálkozzunk a másik (majd a többi) bemenettel! Ezután küldjünk ki hosszú adatsorozatot, pl. bináris 01111000-t (hexa 78, az "x" kódja)! Voltmérővel mérve a kimeneteket, az egyik feszültsége a minimális (0, –5 vagy –12V), és a maximális (+5, +12 V) között lesz valahol, és ha vége az adásnak, visszatér álatlában a minimális értékre, míg a másik valamelyik szélső értéken marad. Az előbbi láb az adatkimenet (TxD), az utóbbi a vevő KESZ jele (DSR vagy RTS). A fennmaradt egy bemenet pedig az adatbemenet (RxD). Két gépet ezek után úgy kapcsolhatunk össze, hogy az egyik adatbemenetét összekötjük a másik adatkimenetével, a DSR lábat pedig a másik CTS lábával, és fordítva. Már csak az adatformátumot és az átviteli sebességet kell egyformára állítani a megfelelő utasítások segítségével, és kezdődhet az adatátvitel!

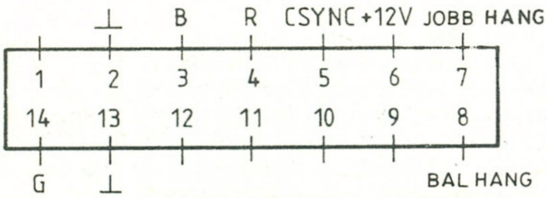
RGB-monitor kimenet: három színjel- és egy szinkronjel-kimenetet kell megtalálnunk. Állítsuk a képernyőt (a keretet és a papírt is) feketére, majd mérjük meg a kivezetések feszültségszintjét! Ezután állítsuk az egész képernyőt pirosra! Újra megmérve a feszültséget, az egyiket sokkal magasabbnak fogjuk találni: ez lesz a piros színjel. Hasonló módon található meg a másik két színjel is. A szinkronjelet próbálgatással kereshetjük az alaphelyzetben 2–4 Voltos kimenetek között.

Nézzük ezek után az eredményeket az Enterprise-on!

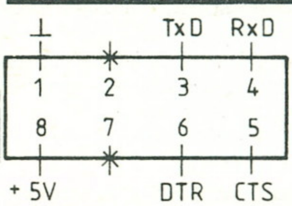
A csatlakozókat hátulnézetben ábrázoljuk, a számozás önkényes. Néhány lábról a fenti rendkívül tudományos módszerrel sem sikerült (egyelőre) kideríteni, hogy micsoda, ezt egy kis vízszintes vonallal jelezzük

MONITOR

1. videó képjel
2. föld
3. kék színjel (B)
4. piros színjel (R)
5. összetett szinkron (RGB-hez)
6. +12 V
7. jobb hangkimenet
8. bal hangkimenet
9. videó összetett szinkronjel
10. TTL képszinkronjel
11. TTL sorszinkronjel
12. összetett fekete-fehér videojel
13. föld
14. zöld színjel (G)



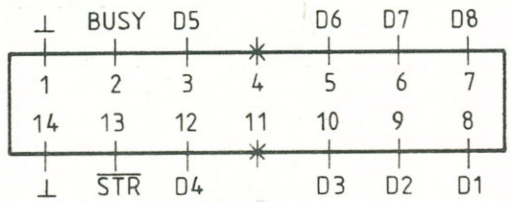
SERIAL/NET



1. föld
2. nincs bekötve
3. TxD
4. RxD
5. CTS
6. DTR vagy RTS
7. nincs bekötve
8. +5 V

PRINTER:

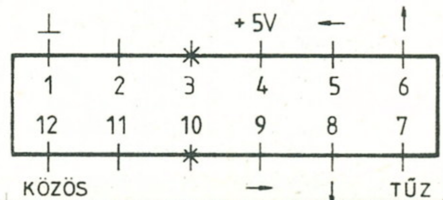
1. föld
2. BUSY bemenet
3. 5. bit
4. nincs bekötve
5. 6. bit
6. 7. bit
7. 8. bit
8. 1. bit
9. 2. bit
10. 3. bit
11. nincs bekötve
12. 4. bit
13. STROBE kimenet
14. föld



CONTROL 1 és 2

1. föld
2. – (kimenet)
3. nincs bekötve
4. +5 V
5. balra
6. fel
7. tűz
8. le
9. jobbra
10. nincs bekötve
11. – (kimenet)
12. joystick közös kivezetés

Mészáros Gyula



800XL ATARI KÖR

Többféle sorozatú Atari 800XL van forgalomban. A különbségük elsősorban a BASIC verzióban mutatkozik meg. Akiknek a gépével programírás közben olykor előfordul, hogy szó nélkül lemerevedik – kiakad – annak föltétlenül érdemes ezt a cikket továbbolvasni. Ilyenkor ugyanis csak a gép kikapcsolásával – azaz az addigi munka elvesztésével – lehet továbblépni, s ez meglehetősen bosszantó. Nos ez a jelenség a gép BASIC-jének hibája. A korábbi gépek (400 és 800) BASIC-je tartalmazott egy csomó hibát, amelyet a 600XL és 800XL megjelenésekor javítani kívántak a BASIC úgynevezett „B verzió"-jával. A hibát jórészt sikerült is kijavítaniuk, ez a fenti hiba azonban megmaradt, sőt egy újabb is „képződött". Nevezetesen az, hogy a SAVE-vel kimentett programok minden egyes mentéskor 16 byte-tal hosszabbak lesznek. A gépcsalád harmadik „generációja" kiadása előtt (65XE és 130XE) újból javítottak a BASIC-en a „C verzió"-val és ez már sikerrel is járt. A 800XL-ek későbbi sorozatai már ezt a BASIC-et tartalmazzák. Hogy kinek milyen – A, B vagy C BASIC-et tartalmazó gépe van, ezt egyszerűen kideríthetjük az alábbi módon:

Közvetlen üzemmódban (tehát sorszám nélkül) gépeljük be a következőket:

? PEEK(43234) ⟨RETURN⟩

Ha a gép 162-vel válaszol „A verzió", ha 96-al „B verzió", ha 234-el „C verzió" van a gépben. (Ezt egyébként a program maga is ellenőrzi!)

Nos, ha 96-ot kapunk, van értelme az itt közölt program begépelésének. Ez ugyanis a „B verzió"-ból „C"-t csinál! A programot futtatás előtt mentsük el, esetleges javítás esetére, majd futtassuk. A futtatáskor a gép ellenőrzi a BASIC verziót és a diszkre egy BASICC elnevezésű file-t ír.

Ezután ha használni akarjuk:
ENTER "D:BASICC" ⟨RETURN⟩
és utána RUN.

(A PAGE 6 c. angol ATARI folyóiratban megjelent cikk nyomán:)

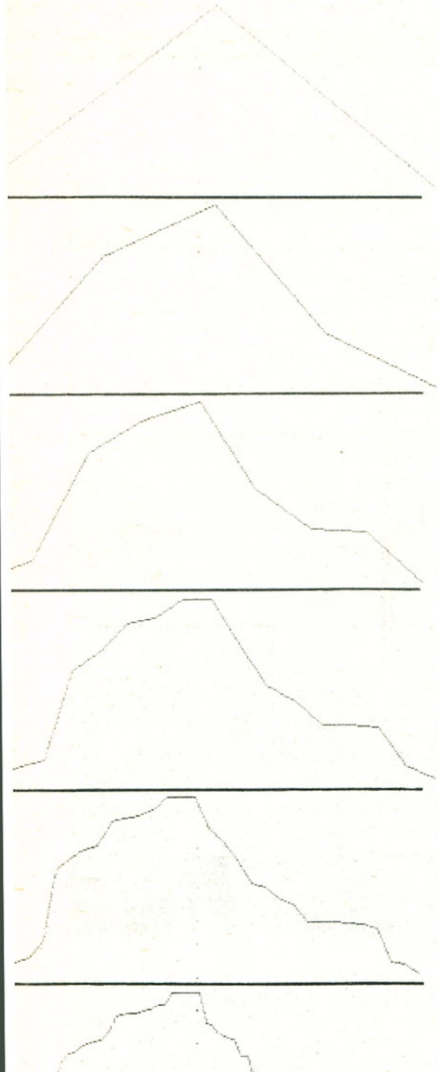
BASICBŐL CASIC

```
30 PRINT CHR$(125);
40 A=PEEK(43234)
50 IF A=162 THEN ? "ELNEZEST ONNEK A VERZIOJU BASIC-JE VAN";:? "EZ SAJNOS NEM AT
ALAKITHATO.":GOTO 380
60 IF A=234 THEN PRINT "ONNEK MAR MEGVAN A C BASIC-JE !":PRINT "ERRE A PROGRAMRA
 SEMMI SZUKSEGE.":GOTO 380
70 PRINT "MOST KESZITEM A ";CHR$(34);"BASICC";CHR$(34);"FILE-T..."
80 OPEN #2,8,0,"D:BASICC"
90 PRINT #2;"10 DIM S$(82)"
100 PRINT #2;"20 S$=";CHR$(34);
110 FOR I=1 TO 82:READ A:PRINT #2;CHR$(A);:NEXT I
120 PRINT #2;CHR$(34)
130 PRINT #2;"30 A=USR(ADR(S$))"
140 PRINT #2;"40 FOR I=1775 TO 1788:READ A:POKE I,A:NEXT I"
150 PRINT #2;"50 DATA 169,255,141,1,211,169,1,133,9,169,0,141,68,2"
153 PRINT #2;"53 P=PEEK(9):IF P/2=INT(P/2) THEN POKE 9,P+1:POKE 1789,96:GOTO 60"
156 PRINT #2;"56 POKE 1789,76:POKE 1790,PEEK(12):POKE 1791,PEEK(13)"
160 PRINT #2;"60 POKE 12,23:POKE 13,6"
170 PRINT #2;"70 PRINT CHR$(125);";CHR$(34);"C VERZIOJU BASIC A RAM-BAN VAN,";CH
R$(34)
180 PRINT #2;"80 PRINT ";CHR$(34);"ES RESET-BIZTOS.";CHR$(34);" : PRINT "
190 PRINT #2;"90 PRINT ";CHR$(34);"B BASICHEZ VISSZA:";CHR$(34);" : PRINT "
200 PRINT #2;"100 PRINT";CHR$(34);"   DOS RETURN";CHR$(34);" : PRINT "
210 PRINT #2;"110 PRINT ";CHR$(34);"      RESET ";CHR$(34);" :PRINT"
230 PRINT #2;"120 END"
240 DATA 104,216,169,0,133,208,169
250 DATA 160,133,209,162,32,160,0
260 DATA 177,208,72,169,255,141,1
270 DATA 211,104,145,208,169,253,141
280 DATA 1,211,136,208,237,230,209
290 DATA 202,208,232,169,255,141,1
300 DATA 211,169,234,141,223,168,141
310 DATA 226,168,169,240,141,224,168
320 DATA 169,17,141,225,168,169,243
330 DATA 133,208,169,191,133,209,169
340 DATA 0,141,41,187,160,6,145
350 DATA 208,136,48,251,96
360 CLOSE #2
370 PRINT "A ";CHR$(34);"BASICC";CHR$(34);" A LEMEZEN VAN"
380 END
```

Sztankay László és Zsolt

hosszú-hosszú vonalak

Iskolai tanulmányaink során megtanultuk és meg is szoktuk, hogy egy-egy vonalszakasznak – legyen az akár egyenes szakasz, akár bármilyen kacskaringós görbe darab – jól meghatározható hosszúsága van. Ez a hosszúság gyakran mérhető, mint az egyenes egy-egy darabjának esetében. Máskor viszont kiszámítható: így például a kör kerületének meghatározásakor is ismert a Ludolf-féle szám – közismert nevén a PI = 3,1415926... – amelynek segítségével tetszőleges pontossággal meg tudjuk adni az adott sugarú körvonal hosszát.



Sok, valamennyire is szabályos görbevonal hosszának számításához léteznek hasonló állandók és képletek, mint a kör esetében – és ettől könnyen elbizakodottá is válhatunk. Elvégre mi is kell egy vonaldarab hosszúságának meghatározásához? Vagy egy jó képlet, ami rögtön szolgáltatja az eredményt, vagy pedig egy pontos mérőeszköz – mondjuk, egy adott hosszúságú **mérőléc** –, amivel lépésről lépésre lemérjük a részhosszúságokat, majd összegezzük őket. Nos, éppen az utóbbi módszert követve kerülünk bajba – ami pedig az egyenes szakaszok esetében nagy hasznunkra volt.

FÉLSZIGET

A legáltalánosabb görbéket vizsgálva könnyen arra a következtetésre juthatunk, hogy a vizsgált vonalszakasz egész egyszerűen „mérhetetlenül" hosszú – a szó szoros értelmében is, azaz nem tudunk olyan mérőlécet találni hozzá, amellyel valóban meg tudnánk határozni a hosszát.

Erre a klasszikus példa egy félsziget, amelynek a partvonal-hosszúságát szeretnénk megmérni. Nosza, **vegyük elő a leghosszabb mércéinket, és fektessük végig a félsziget partján!** Ha ezek hosszait összeadjuk, akkor nyilván lesz már valami **közelítő** adatunk a minket érdeklő mennyiségről. Ha azonban feleakkora mérőrudakat veszünk elő, és ezekkel próbálkozunk, akkor a part hossza **nagyobbnak** fog adódni, mint korábban. Hiszen ezekkel a rövidebb mérőpálcákkal már olyan kisebb öblöcskék, kisebb kiszögellések kerületét is nyomon követhetjük, amelyeket az előbb vaskos botjaink érzéketlenül áthidaltak. Csökketsük most ismét felére a mércék hosszát! Az előbbi okok miatt most újra csak nagyobb eredményt kapunk. És ez így megy, a partszakaszt alkotó homokszemek méretéig – vagy azon is túl... Ezt próbálja érzékeltetni első programunk, mely – mint a többi bemutató program is – Plus/4 gépre készült.

A program egy első látásra derékszögű háromszöghöz hasonlító félszigetet (vagy tengeröblöt – kinek-kinek ízlése szerint) térképez fel, mind jobban finomítva a mérés pontosságát. Ennek végül is persze határt szab a képernyő, illetve a számítógép felbontotóképessége – és ez érvényes a további programjainkra is. Vagyis a későbbiekben hiába hivatkozunk időnként a „végtelenre", **a programok csak addig futnak, amíg látható a képernyőn valami változás.** Ennek ellenére hisszük, hogy a látvány nyújt majd valami újdonságot a programok bepötyögőinek.

```
5 REM ***** FELSZIGET *****
10 COLOR0,2:COLOR1,1
20 GRAPHIC2,1
30 DIMA(256)
40 A(0)=128:A(128)=0:A(256)=128
50 FORK=7TO0STEP-1
60 LOCATE32,A(0)+20
70 FOR I=2↑KTO256STEP2↑K
80 DRAW1TOI+32,A(I)+20
90 A(I-2↑(K-1))=(A(I)-A(I-2↑K))*RND(1)
100 A(I-2↑(K-1))=A(I-2↑(K-1))+A(I-2↑K)
110 NEXTI:GETKEYA$:SCNCLR: NEXTK
120 GRAPHIC0
```

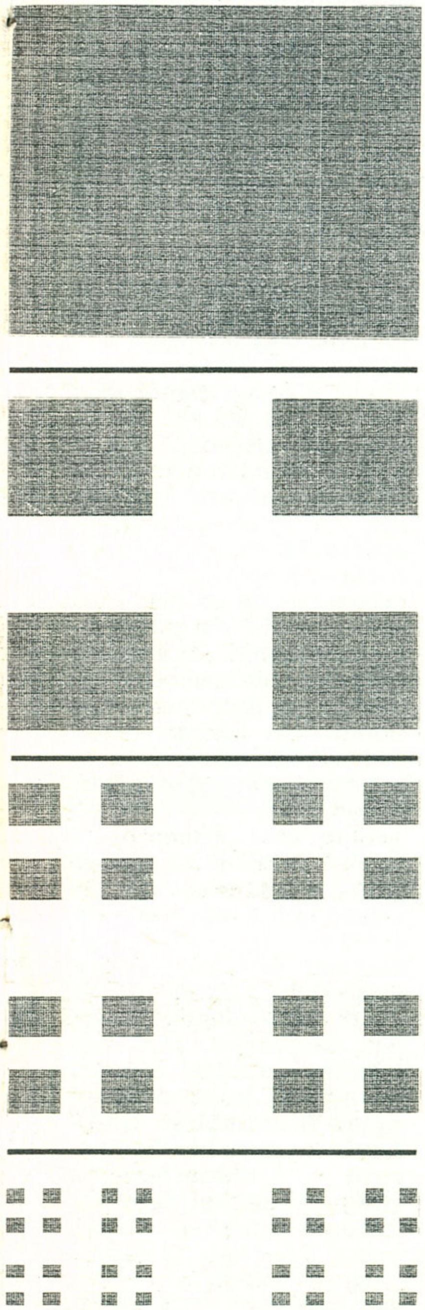
PONTOK TERÜLETE

A vonalról kezdtünk beszélni, és itt is folytatjuk. **Tulajdonképpen mit is nevezünk vonalnak?** Euklidesz – a ma közkeletűen használt, már az általános iskolában is tanított geometria megalapozója – egyik művében az egyenest mint **„szélesség nélküli hosszúság"**-ot említi. Ezt így tanultuk, így tudjuk. Azt is tudjuk, hogy a pont az a geometriai alakzat, amelynek sem szélessége, sem hosszúsága – azaz semmilyen irányú kiterjedése – sincsen. A következő példa azonban talán elgondolkodásra késztet mind a pont, mind a vonal fogalmával kapcsolatban:

Rajzoljunk egy egységnyi oldalhosszúságú – és így egységnyi területű – négyzetet, majd vágjunk ki belőle egy nagy keresztet úgy, hogy még mindig megmaradjon a négyzet 3/4 része – vagyis a kereszt területe legyen 1/4. Négy kis négyzetünk maradt, de ezeket csonkítsuk meg ismét egy-egy kereszttel, úgy,

hogy a most elvett keresztek összterülete legyen 1/8. A megmaradó négyzetekből vegyünk el ismét kereszt alakú részeket, amelyeknek összege most 1/16 legyen – és folytassuk tovább ezt a nem túl tisztességes „négyzetcsonkító" eljárást akár a végtelenségig. Nyilván mindenki úgy gondolja, hogy ha nagyon sokáig műveljük ezt a négyzetekkel, akkor azok ponttá zsugorodnak össze, így nulla lesz a területük. Nos, egyenként, egy-egy ilyen négyzetet vizsgálva ez igaz is lehet – de a négyzetek (pontocskák) összességét illetően semmiképp sem. Ugyanis először eltávolítottuk a teljes négyzet területének 1/4 részét. Maradt 1–1/4. Következett az 1/8 rész törlése – de ezután is megmaradt a teljes négyzet 1–(1/4+1/8) része. Nem jártunk sokkal jobban az újabb kereszt letörlésével sem: még így is megmaradt a teljes négyzet 1–(1/4+1/8+16) része. Még aki nem is ért a határétékszámításhoz, nyilván az is rögtön rájön, hogy a zárójelben lévő kifejezés soha nem éri el az egyet azaz négyzetünk feltételezett területét. Aki viszont valamelyest is konyít a matematikának ehhez az ágához, az tudja, hogy amit zárójelbe tettünk, az végtelen sok lépés után is legfeljebb 1/2 lehet – így **az eredeti négyzetből még a sok-sok kereszt után is olyan „pontok" együttese marad, amelyek összterülete megegyezik az eredeti négyzet felével.**

Vagyis e pontoknak területük van, amely egyenlő a teljes négyzet területének felével!



VONAL TERÜLETE

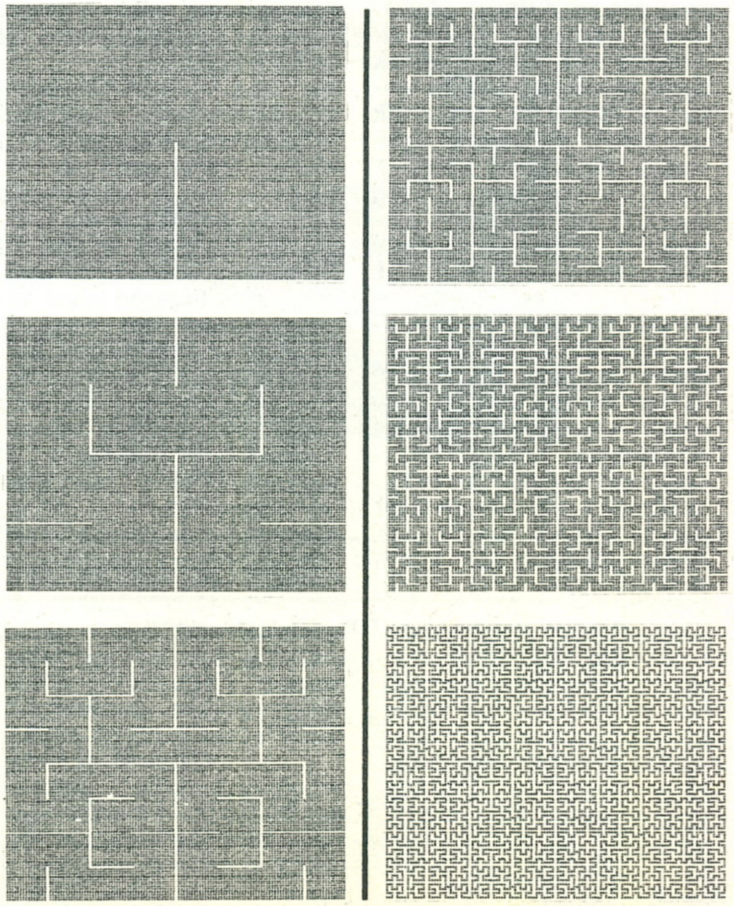
Most talán már kellőképpen sikerült elbizonytalanodnunk a „szélesség", „hosszúság" fogalmakat illetően. De hogy egy vonalnak is lehessen területe – vagy legalábbis afféle, amit mi annak nevezünk?

Tegyük a következőt! Az egyszerűség kedvéért válasszunk megint egy négyzetet, majd ezt osszuk fel úgy, hogy egy „U" alakú sáv mentén be lehessen járni. Ez nem nehéz, hiszen csak a négyzet középpontjától kell egy egy egyenes szakaszt húznunk valamelyik oldalfelezőpontig, és a sáv máris előáll. A második lépés már nehezebb – az imént felosztott területet tovább bontani úgy, hogy a fekete sáv összefüggő maradjon, azaz sehol ne szakadjon meg.

Minden további magyarázkodás helyett jobb, ha megnézzük a Peano görbe programot, illetve futtatásának eredményét!

A kapott ábrából látható, hogy a **fekete sáv egyre**

```
5 REM ***** PEANO-GORBE *****
10 COLOR0,2
20 COLOR1,1
30 GRAPHIC2,1
40 BOX1,97,31,223,157,,1
50 DRAW0,160,94TO160,158
60 FORK=0TO4
65 R=2↑(5-K)
70 FORI=1TO2↑K
80 FORJ=1TO2↑K
90 LOCATE96+(I*2-1)*2*R,30+(J*2-1)*2*R
120 LOCATE+R,+0
125 IFRDOT(2)=0THENX=1:LOCATE-R,+0:GOTO160
130 LOCATE-2*R,+0
135 IFRDOT(2)=0THENX=2:LOCATE+R,+0:GOTO160
140 LOCATE+R,+R
145 IFRDOT(2)=0THENX=3:LOCATE+0,-R:GOTO160
150 LOCATE+0,-R*2
155 IFRDOT(2)=0THENX=4:LOCATE+0,+R:GOTO160
160 ONXGOSUB200,300,400,500
170 NEXTJ:NEXTI:NEXTK
180 END
200 LOCATE-R,+R
210 DRAW0TO+R,+0TO+0,-2*RTO-R,+0
220 LOCATE+0,+R:DRAW0TO-R,+0
225 LOCATE+2*R,+2*R:DRAW0TO+R,+0TO+0,-R
230 LOCATE+R,+0:DRAW1,+0,-2*R
240 LOCATE-R,+0:DRAW0TO+0,-RTO-R,+0
250 RETURN
300 LOCATE+R,-R
310 DRAW0TO-R,+0TO+0,+2*RTO+R,+0
320 LOCATE+0,-R:DRAW0TO+R,+0
325 LOCATE-2*R,-2*R:DRAW0TO-R,+0TO+0,+R
330 LOCATE-R,+0:DRAW0TO+0,+2*R
340 LOCATE+R,+0:DRAW0TO+0,+RTO+R,+0
350 RETURN
400 LOCATE -R,-R
410 DRAW0TO+0,+RTO+2*R,+0TO+0,-R
420 LOCATE-R,+0:DRAW0TO+0,-R
425 LOCATE-2*R,+2*R:DRAW0TO+0,+RTO+R,+0
430 LOCATE+0,+R:DRAW0TO+2*R,+0
440 LOCATE+0,-R:DRAW0TO+R,+0TO+0,-R
450 RETURN
500 LOCATE+R,+R
510 DRAW0TO+0,-RTO-2*R,+0TO+0,+R
520 LOCATE +R,+0:DRAW0TO+0,+R
525 LOCATE+2*R,-2*R:DRAW0TO+0,-RTO-R,+0
530 LOCATE+0,-R:DRAW0TO-2*R,+0
540 LOCATE+0,+R: DRAW0TO-R,+0TO+0,+R
550 RETURN
```



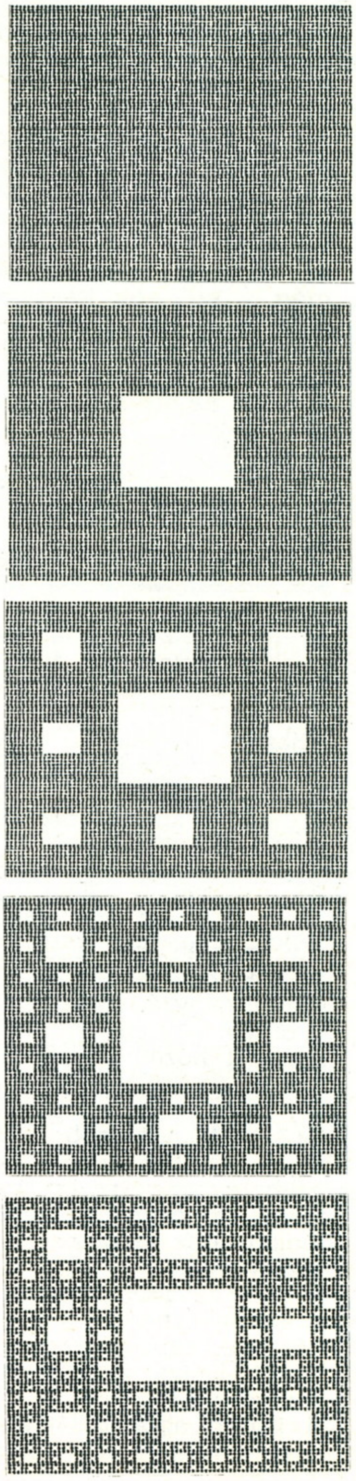
hosszú-hosszú vonalak

inkább elvékonyodik, de eközben mindvégig összefüggő marad. Ugyanakkor kiderül az is, hogy az eredeti négyzetnek mind több és több pontján halad át. Magyarul: ha ezt a sávszűkítést, görbítgetést minden határon túl folytatjuk, akkor a minket érdeklő sáv **vonallá** szűkül össze. Ez a vonal viszont átmegy a négyzet minden pontján. Azaz területe lesz, ami megegyezik az eredeti négyzet területével.

SZIGET ÉS TAVAK

Ezek után úgy tűnik, hogy ismét meg kell vizsgálnunk az egyenes iménti, Euklidesz által adott meghatározását. Nos, Euklidesz még egy meghatározását adta a vonalnak: egy másik leírása szerint a vonal **„felületek határa"**. Első pillantásra ezt is érthetőnek tartja mindenki: vagyis vonal (szakasz) az, ami a sík két részét elválasztja egymástól. Az igazság azonban az, hogy ez sincs mindig így. Ezt mutatja meg következő példánk, amely Wada japán grafikustól ered.
Így szól a mese: egy szigetet minden oldalról tengervíz vesz körül. Ezen a szigeten van egy hideg vizű és egy meleg vizű tó. Egy napon a sziget kormányzója elhatározza, hogy a lakosságot ellátja mindhárom fajta vízzel: a sós tengervízzel, a meleg és a hideg vízzel is. Aznap csatornát ásat a tengerpartról és a két tó partjáról úgy, hogy azok ne keresztezzék egymást – elvégre akkor összekeverednének a különböző minőségű vizek –, de azt is megkívánja munkásaitól, hogy a sziget bármely pontjától egyik csatorna sem legyen távolabb, mint egy hosszúságegység (mondjuk 1 km, hogyha kis szigetről van szó).
Eljön a következő nap, és a kormányzó még többet akar tenni népéért. Utasítja munkásait, hogy folytassák a három csatorna kialakítását úgy, hogy azok bárhonnan, legfeljebb fél kilométeres távolság megtételével elérhetőek legyenek. Azután a kormányzó napról napra úgy adja ki utasításait, hogy a csatornák fele akkora távolságra legyenek a sziget bármely pontjától, mint az előző napon.
Ha most feltételezzük, hogy a kormányzó örökéletű – hiszen meséről van szó –, akkor nyilvánvaló az is, hogy a sziget földterülete **vonalvékonyságúra** finomodik. S végül is újra itt áll előttünk egy vonal, amihez tetszőlegesen közel találunk tengervizet, hideg tóvizet és meleg tóvizet is – azaz **a szigetből megmaradt vonalszerű földsáv nem két, hanem három tartományt választ el egymástól:** a háromféle víztartományt. Egy ehhez nagyon hasonló vonalat állít elő következő programunk, amelyet kitalálójáról, a lengyel matematikusról általában „Cantor-terítőnek" neveznek.

```
5 REM ***** CANTOR-TERITO *****
10 COLOR0,2
20 COLOR1,1
30 GRAPHIC 2,1
31 A=120:B=60
32 BOX1,A,B,81+A,81+B,,1
40 FORK=0TO 3
50 FORI=1TO3↑K
60 FORJ=1TO3↑K
65 P=I*3-2:Q=J*3-2:F=3↑(3-K)
67 X0=A+P*F:Y0=B+Q*F
70 BOX0,X0,Y0,X0+F-1,Y0+F-1,,1
80 NEXTJ:NEXTI:GETKEYA$:NEXTK
90 GRAPHIC0
```

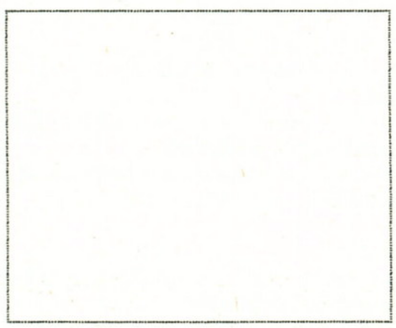


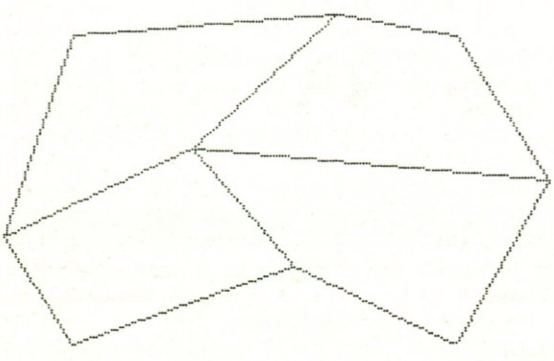
Rajzoljunk újra egy feltöltött négyzetet! Ezután osszuk fel 3*3 – azaz 9 – kisebb négyzetre, majd távolítsuk el ezek közül a középsőt! A megmaradó keretet alkotó nyolc kisebb négyzet mindegyikét osszuk újra kilenc-kilenc részre, és ezek közül is töröljük mindig a középsőt!
Talán már mondanunk sem kell, hogy ezt az eljárást is a „végtelenségig" kell folytatnunk, míg a megmaradó hálózat vonallá finomodik. Végül is az így megmaradó vonalnak is területe lesz – mégpedig elég nagy (ha ez a kifejezés használható itt): az eredeti négyzet területének 7/8 része. Ez a görbe egyébként még számos érdekes tulajdonsággal rendelkezik – de ez már kívül esik mostani vizsgálódásaink tárgykörén. Az érdeklődőknek a halmazelméleti topológia, és azon belül is a Cantor-féle görbék tanulmányozását ajánljuk.

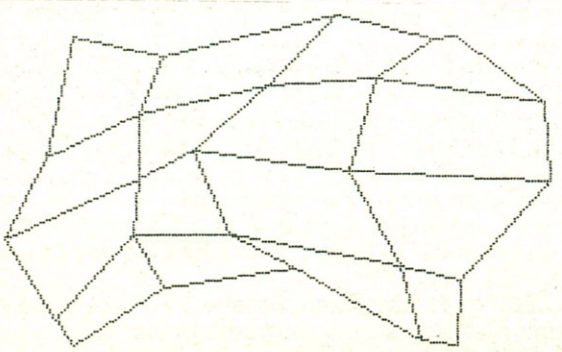
ZSEBKENDŐ

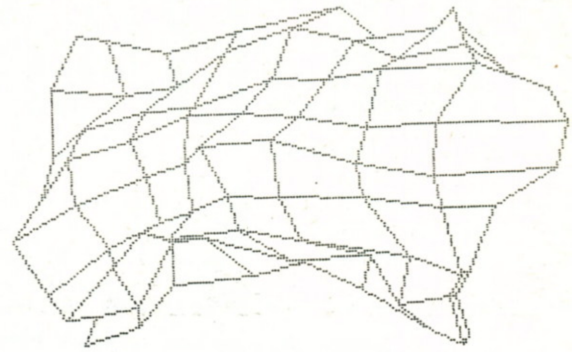
Térjünk vissza inkább a már az első példánkban említett félsziget-partvonalhoz!
Ugyanúgy, ahogy egy-egy, akármennyire görbült vonalszakaszt tetszőleges pontossággal közelíthetünk mindinkább rövidülő egyenes darabokkal, úgy **egy dimbes-dombos felszínt is egyre pontosabban, egyre finomabban lefedhetünk egyre kisebb méretű síkdarabokkal (négyszögekkel vagy háromszögekkel).** Így, ahogy egy félsziget partvonalának hosszát fokozatosan csökkenő hosszúságú mérőlécekkel közelítettük, ugyanígy például egy hegyorom felszínét is megmérhetjük mind kisebb négyzetlapokkal.
Ezt a módszert egyébként tetszőleges, elképzelt felületek megrajzolására is

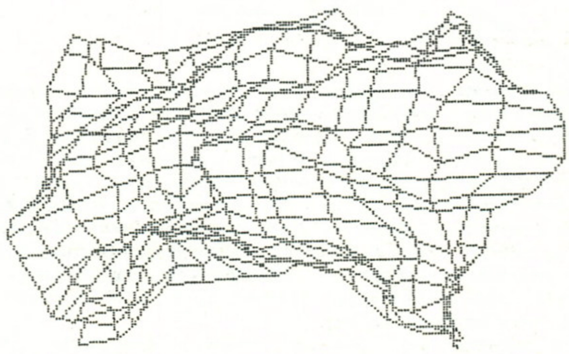
```
5 REM ***** ZSEBKENDO *****
10 P=64:DIMA(P,P,1)
15 GRAPHIC1,1
20 A(0,0,0)=96:A(0,P,0)=224
30 A(P,0,0)=96:A(P,P,0)=224
40 A(0,0,1)=36:A(0,P,1)=36
50 A(P,0,1)=164:A(P,P,1)=164
110 FORK=0TO6
120 L=P/(2↑K)
130 FORI=0TOPSTEPL
140 A=A(I,0,0):C=A(I,0,1)
150 FORJ=LTOPSTEPL
160 B=A(I,J,0):D=A(I,J,1)
170 DRAW1,A,CTOB,D
175 A(I,J-L/2,0)=(A+B)/2+L*(RND(1)-.5)
176 A(I,J-L/2,1)=(C+D)/2+L*(RND(1)-.5)
180 A=B :C=D
190 NEXTJ:NEXTI
200 FORJ=0TO64STEPL
210 A=A(0,J,0):C=A(0,J,1)
220 FORI=L TOPSTEPL
230 B=A(I,J,0):D=A(I,J,1)
240 DRAW1,A,CTOB,D
242 A(I-L/2,J,0)=(A+B)/2+L*(RND(1)-.5)
243 A(I-L/2,J,1)=(C+D)/2+L*(RND(1)-.5)
245 IFJ<>0THEN GOSUB600
320 A=B:C=D
330 NEXTI:NEXTJ
332 GETKEYA$
335 SCNCLR
340 NEXTK
500 GRAPHIC0,0:END
600 Q=A(I,J,0)+A(I,J-L,0)+A(I-L,J,0)
610 Q=(Q+A(I-L,J-L,0))/4+L*(RND(0)-.5)
630 A(I-L/2,J-L/2,0)=Q
640 Q=A(I,J,1)+A(I,J-L,1)+A(I-L,J,1)
650 Q=(Q+A(I-L,J-L,1))/4+L*(RND(0)-.5)
670 A(I-L/2,J-L/2,1)=Q
680 RETURN
```

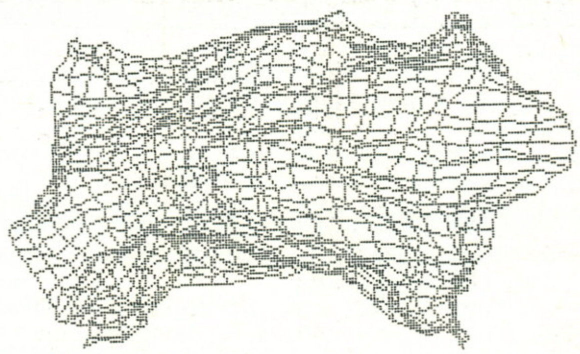


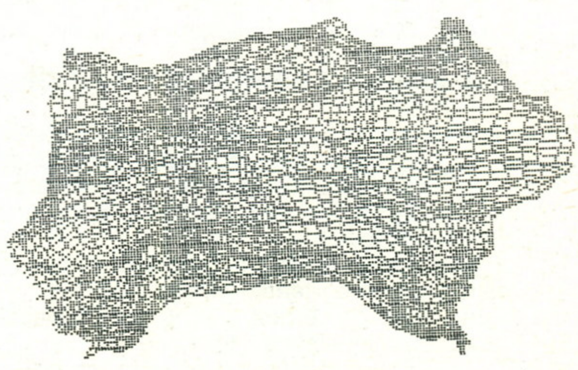












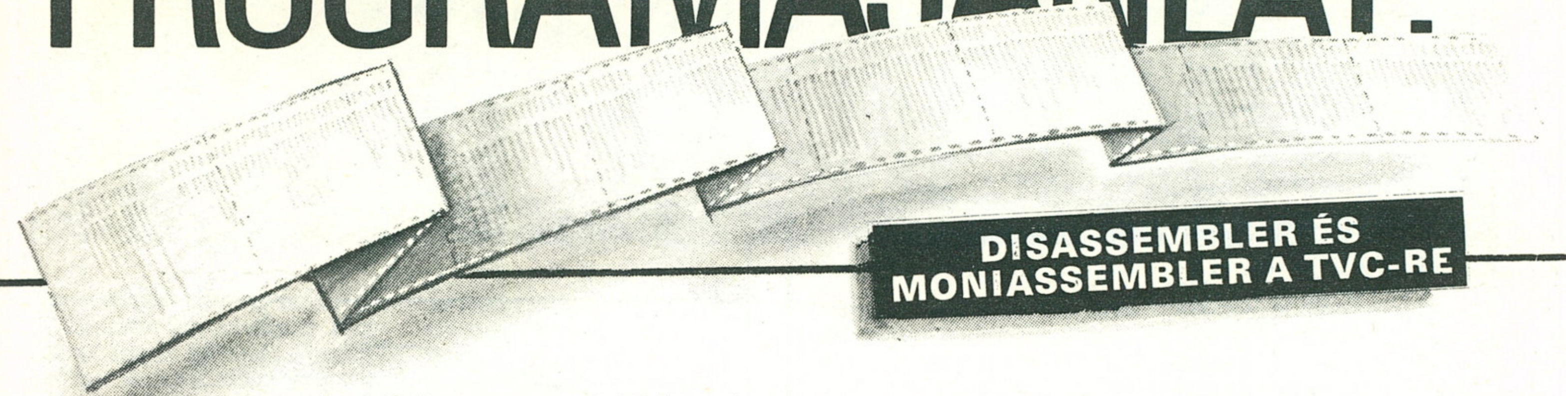
használhatjuk, és bizonyára az olvasó is találkozott már ezzel – bár talán nem tud róla: így például a „Csillagok Háborúja" sorozat háttereinek megtervezésénél is gyakran ezt a módszert alkalmazták.

Utolsó példaprogramunk egy efféle felület kialakulását mutatja be. Szokásunkhoz híven ismét egy négyzetből indulunk ki, azután ennek oldalhosszúságát felezzük több lépésben. Ez persze még nem eredményezne térbeli hatást, de az újonnan kialakuló résznégyzetek középpontját, valamint oldalfelező pontjait mindig eltoljuk egy véletlen értékkel. Így azután a kapott kép olyan lesz, mintha egy zsebkendőt folyamatosan összegyűrnénk.

Tallér József

ötlet

PROGRAMAJÁNLAT:



DISASSEMBLER ÉS MONIASSEMBLER A TVC-RE

Ritka szerencse ha egy lap közül egy programot, közben valaki fejleszt egy másikat, amely sok szempontból kiegészíti azt, s véletlenül a két program össze is illeszthető. Nos, ezúttal ez történt. Májusban jelent meg a BIT-LET-ben Dörner Péter TVC-re készített monitor programja. Ezután érkezett hozzánk Szoldatics József disassemblere. Akinek tehát sikerült a monitort bepötyögnie, annak érdemes ezt a programot is beírni, majd a közölt módon összefésülni a kettőt.- Azt már csak az érdekesség kedvéért jegyezzük meg, hogy az összefésülési (MERGE) eljárást is a BIT-LET-ből vettük (1986. szeptemberi szám).
Természetesen aki önállóan, csak a disassemblert akarja használni, ezt is megteheti.

DISASSEMBLER

A programmal megnézhetjük mi van a memóriában, de sajnos beírni nem lehet vele, viszont tud disassemblálni, és ha szöveg van a memóriában kiírathatjuk vele a karaterek ASCII kódjait.
Fontos tudnivaló, hogy mivel a program BASIC-ben íródott, a ROM-területet nem lehet olvasni vele!

KEZELÉSI ÚTMUTATÓ

A program beolvasása után a gépet nagybetűs üzemmódba kell állítani. **(CTRL+LOCK)**
A RUN parancs kiadása után a program a következőkkel jelentkezik be: paging byte 70H, fordítási cím 0000H, nyomtató kikapcsolva és Disassembler üzemmód. Ezután a rendszer parancsra vár, ami egy billentyű megnyomását jelenti. A parancs billentyű funkciójának végrehajtása után a rendszer újra parancsra vár.
A parancsok kiadása a következő billentyűkkel történik:
N – Megváltozik a nyomtató állapota, mely a képernyőn is követhető. A "be" felirat azt jelzi, hogy mindaz az információ, ami a képernyőn megjelenik, a nyomtatón is kinyomtatódik. A „ki" felirat pedig azt jelzi, hogy a program csak a képrnyőre dolgozik.
A – Új fordítási címet kérdez a program. A cím beadása hexadecimális alakban történik. (nagybetűk!!)
P – Új paging byte-ot kérdez a program. A paging byte beadása hexadecimális alakban történik.
„Szóköz" – A képernyőn (nyomtatón is, ha be van kapcsolva!) egy üres sor jelenik meg (fordítások tördelése!).

Ezek a billentyűk nem (!) változtatják meg a program üzemmódját. Más üzemmódot a következő billentyűkkel lehet elérni.
D – Disassembler üzemmód. Ekkor a gép a beállított címről fordít, fordítás után a címmutatót (Address) lépteti.
W – Word üzemmód. Ekkor a címtől egy kétbyte-os számot olvas, a gépi nyelvhez hasonlóan fordított sorrendben.
B – Byte üzemmód. Először megkérdezi, hogy mennyi byte-ot írjon egy sorba (min. 1, max. 12), majd a parancsot végrehajtja.
T – Text üzemmód. Ekkor ASCII kódokat ír ki (már ha van, ha nincs, akkor "." jelenik meg helyette). Először itt is a byte-ok számát kell beállítani (min. 1, max. 24).

Minden, ezektől különböző billentyű megnyomása esetén a beállított üzem egyszeri végrehajtása történik, de a Byte és Text üzemmódban nem kéri a byte-szám beállítását újra.
A képernyőn megjelenő számok mind hexadecimálisak!

PROGRAM LEÍRÁSA

3010–3150: Kezdőértékek felvétele, ábra rajzolása, alapállásba állás
3160–3210: Főág. Mindig ide tér vissza, itt várja a billentyű megnyomását, majd meghívja a végrehajtó szubrutint, és újra vár.
3500–3950: üzemmódok szubrutinjai.
5000–5530: A fordító szubrutin magja, már magában is többé-kevésbé működőképes.
6000–6340: Közhasznú szubrutinok, több program is használja őket.
8000–8700: A működéshez szükséges DATA-k.
Néhány kitüntetett változó:
LÉPÉS$: tartalmazza a parancsbillentyűket
MOD: üzemmód parancsbillentyűjének sorszáma LÉPÉS$-ben
PC: fordítási cím
VS$: minden kiíratandó információ a V$-be kerül, és ez íratódik ki a megfelelő helyre
PAGING: tartalmazza a Paging byte értékét
NY: nyomtató állapotát tartalmazza,
ha 1, akkor kikapcsolva
ha 2, akkor be-kapcsolva a nyomtató
A program bővítésének módja:
Az új parancs betűjének beírása a LÉPÉS$-be a 3150-es sorba, majd a 3200-as sorba a parancs végrehajtásának sorszámát kell beírni. A végrehajtási rutin végén RETURN-nek kell lenni.
Szoldatics József, Kapuvár, Gimnázium 9330.

ÖSSZEFÉSÜLÉS

Most jöhet – ha akarjuk – a két program összefésülése.
Éspedig a következőképpen:
1. Betöltjük vagy beírjuk a májusban megjelent monitor programot.
2. A 420-as sorban található 1E4 helyett (10000) 16500-at írunk.
3. Kibővítjük a következő sorral:
1065 IF QW$="a" THEN POKE 2918,1:RUN 3010
4. Ezután a programból **kazettás file**-t csinálunk:
OPEN OUTPUT"MONITOR":LLIST#5:1–2950: CLOSE OUTPUT
Sorzárás előtt indítsuk el a magnót!
5. Ezután **betöltjük** vagy beírjuk a disassemblert.
6. A program 3150-es sorában a **LÉPÉS$ végét kiegészítjük** egy M-mel
7. A **3200 sort kiegészítjük** egy 3300-as sorszámmal.
8. Írunk még egy sort a programhoz, éspedig:
3300 POKE 2918,0:RUN

9. Ha ez kész, akkor kiadjuk az alábbi parancsot és megvárjuk amíg a két program „összefésülködik"

OPEN"MONITOR":POKE 2818,5:CLOSE

(a magnót természetesen az előbb készített kazettás file-hoz tekerjük és elindítjuk)

Vigyázat! A beillesztés művelete hosszabb ideig tart mint a kazettán két program közti szünet. Ezért a nem távvezérléses magnókat a DISASSEMBLER végén állítsuk meg! Ha a dolog így nem működik, akkor az OPEN utasítást külön kell kiadni, s a továbbiakat azután, hogy a MONITOR fejét betöltöttük.

10. Végül az összefésült programokat (programot) **mentsük** kazettára.

Az összefésült MONIASSEMBLER használata:

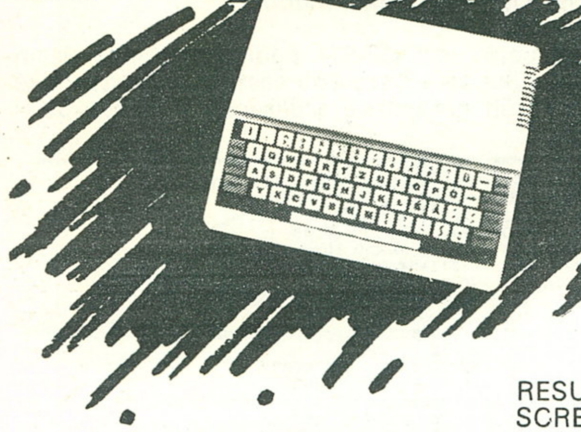
Programindítás után a **monitor** jelentkezik be. A disassemblerbe az **"a"** és a RETURN hatására léphetünk. Visszafelé pedig az **"M"** billentyű megnyomásával ugorhatunk, ekkor a monitor elölről indul.

Figyelem!

A Monitor kisbetűs, a DISASM pedig nagybetűs üzemmódban működik. Az egyszerűbb használhatóság érdekében az összefésült program a szükséges átállításokat elvégzi.

```
3010 GRAPHICS 4
3020 SET PALETTE 81,0,65,20;BORD: 17
3030 TINTA=1:PAPIR=0
3040 SET INK TINTA;PAPER 3
3050 PRINT AT 1,3:STRING$(28,126
3060 PRINT AT 2,3:"* TVC Z-80 DISM (C) Sz.J *"
3070 PRINT AT 3,3:STRING$(28,126
3080 SET PAPER PAPIR
3090 PLOT 32,662;991,662,32,658;1,658
3100 PC=0:PAGING=112:NY=1:MOD=1
3110 GOSUB 6150:GOSUB 6170
3120 GOSUB 6190:GOSUB 6230
3130 GOSUB 6050
3140 DIM OP$(39)*8,V$*45,Q$*32,Q*32
3150 LEPES$="DANP WBBTT":M=1
3160 GET X$
3170 FOR I=1 TO LEN(LEPES$)
3180 IF X$=LEPES$(I) THEN M=I:GO 3200
3190 NEXT I:M=MOD
3200 ON M GOSUB 3590,3570,3500,30,3510,3700,3760,3780,3860,3880
3210 GOTO 3160
3500 NY=((NY-1) XOR 1)+1:GOTO 62
3510 ON NY GOTO 3530,3520
3520 LPRINT
3530 PRINT AT 9,1:CHR$(25):RETUR
3540 PRINT AT 7,19:"";:INPUT PRCT "Paging=":Q1$:Q1$=Q1$(:2)
3550 GOSUB 6110:PAGING=Q1:GOSUB 90:GOSUB 6180
3560 PRINT AT 7,19:STRING$(12,32RETURN
3570 PRINT AT 7,19:"";:INPUT PRCT "Address=":Q$:Q$=Q$(:4)
3580 GOSUB 6130:PC=Q:GOSUB 6160:TO 3560
3590 MOD=M:GOSUB 6230:P_C=PC:GOS 5000:GOSUB 3530:GOSUB 6330
3600 PC=PC+L*INT(A/140)+INT((A-I(A/140)*140)/35)
3610 FOR I=P_C TO PC
3620 CIM=I:GOSUB 6070:W1=ADAT:GOB 6010:V$(7+2*(I-P_C):8+2*(I-P_C))=W1$
3630 GOSUB 6290:V$(17+I-P_C)=CHRADAT)
3640 NEXT I
3650 ON NY GOTO 3670,3660
3660 LPRINT V$
3670 PRINT V$(:5)&V$(7:15)&V$(272)
3680 PC=PC+1:GOSUB 6160
3690 RETURN
3700 MOD=M:GOSUB 6230
3710 GOSUB 6340
3720 W=PC:GOSUB 6030:V$(:4)=W$
3730 CIM=PC:GOSUB 6070:W=ADAT
3740 CIM=PC+1:GOSUB 6070:W=W+256DAT:GOSUB 6030:V$(32:35)=W$
3750 V$(27:28)="DW":P_C=PC:PC=PC:GOSUB 3530:GOSUB 6330:GOTO 3610
3760 MOD=M+1:GOSUB 6230
3770 GOSUB 6310:X=INT(X):IF X<1R X>12 THEN 3770
3780 PRINT AT 4,18:"";:PRINT USE "##":X:GOSUB 6340:FOR I=1 TO X
3790 CIM=PC+I-1:GOSUB 6070:W1=AD:GOSUB 6010
3800 V$(6+2*(I-1):7+2*(I-1))=W1$
3810 NEXT I
3820 W=PC:GOSUB 6030:V$(:4)=W$:PC+X:GOSUB 6160:GOSUB 3530
3830 GOSUB 6330:ON NY GOTO 3850340
3840 LPRINT V$
3850 PRINT V$(:30):RETURN
3860 MOD=M+1:GOSUB 6230
3870 GOSUB 6310:X=INT(X):IF X<1R X>24 THEN 3870
3880 GOSUB 6340:PRINT AT 4,18:":PRINT USING "##":X
3890 FOR I=1 TO X
3900 CIM=PC+I-1:GOSUB 6070:GOSUB6290:V$(5+I)=CHR$(ADAT)
3910 NEXT I:GOSUB 3530:W=PC:GOS 6030:V$(:4)=W$:PC=PC+X
3920 ON NY GOTO 3940,3930
3930 LPRINT V$
3940 GOSUB 6330:PRINT V$(:30)
3950 GOTO 6160
5000 RESTORE 8000:FOR I=1 TO 39EAD OP$(I):NEXT I
5010 D=0:L=0:A=0:GOSUB 6340
5020 CIM=PC:GOSUB 6070:X=ADAT:WC:GOSUB 6030:V$(:4)=W$
5030 CIM=PC+L+1:GOSUB 6070:W1=AT:GOSUB 6010:OP$(21)=W1$
5040 OP$(17)(2:3)=OP$(21)
5050 CIM=PC+1:GOSUB 6070:W=PC+2DAT+256*(ADAT>127):GOSUB 6030
5060 OP$(19)=W$:CIM=PC+2:GOSUB 70:W=256*ADAT
5070 CIM=PC+1:GOSUB 6070:W=W+AD:GOSUB 6030:OP$(20)=W$
5080 OP$(18)(2:5)=W$:IF D=1 THE5330
5090 ON X/64+1 GOTO  5100,5200,10,5120
5100 RESTORE 8100:GOTO 5130
5110 RESTORE 8200:GOTO 5130
5120 RESTORE 8300
5130 FOR I=0 TO X AND 63:READ A:NEXT I
5140 IF A+B=0 THEN 5230
5145 RESTORE 8400:IF (X AND 7)=THEN W1=(X AND 56):GOSUB 6010:OP$(17)=W1$
5150 C=A-35*INT(A/35):FOR I=1 TC:READ C$:NEXT I
5160 C$=C$&STRING$(5-LEN(C$),32V$(27:31)=C$
5170 C=B-40*INT(B/40):IF C=0 TH RETURN
5180 X$=OP$(C):C=INT(B/40):IF C0 THEN X$=X$&","&OP$(C)
5190 V$(32:)=X$&STRING$(12-LEN(),32):RETURN
5200 IF X=118 THEN V$(27:30)="ET":RETURN
5210 V$(27:28)="LD":X$=OP$((X A 56)/8+1)&","&OP$((X AND 7)+1)
5220 A=140:GOTO 5190
5230 IF X=237 THEN 5310
5240 IF X=203 THEN 5470
5250 IF X=221 THEN OP$(7)="(IX+)":OP$(11)="IX":OP$(38)="(IX)"
5260 IF X=253 THEN OP$(7)="(IY+)":OP$(11)="IY":OP$(38)="(IY)"
5270 CIM=PC+2:GOSUB 6070:OP$(7)(=CHR$(43-2*(ADAT>127))
5280 W1=(ADAT>127)*256+ADAT:W1=S(W1):GOSUB 6010
5290 OP$(7)(5:6)=W1$:L=1:PC=PC+CIM=PC:GOSUB 6070
5300 X=ADAT:GOTO 5030
5310 PC=PC+1:CIM=PC:GOSUB 6070=ADAT:D=1:L=0
5320 OP$(11)="HL":OP$(7)="(HL)GOTO 5030
5330 D=0:IF ((X XOR 66) AND 20<>0 THEN 5350
5340 V$(27:29)="SBC":B=371+5*(AND 48)/2:GOTO 5170
5350 IF ((X XOR 74) AND 207)<>THEN 5370
5360 V$(27:29)="ADC":B=371+5*(AND 48)/2:GOTO 5170
5370 IF ((X XOR 64) AND 199)<>THEN 5390
5380 V$(27:28)="IN":B=681+(X A 56)/8:OP$(17)="(C)":GOTO 5170
5390 IF ((X XOR 65) AND 199)<>THEN 5410
5400 V$(27:29)="OUT":B=57+5*(XND 56):OP$(17)="(C)":GOTO 5170
5410 IF ((X XOR 67) AND 207)<>THEN 5430
5420 V$(27:28)="LD":B=378+5*(XND 48)/2:PC=PC+2:GOTO 5170
5430 IF ((X XOR 75) AND 207)<>THEN 5450
5440 V$(27:28)="LD":B=729+(X A 48)/16:PC=PC+2:GOTO 5170
5450 RESTORE 8500
5460 READ C$,C:IF C*X=C*C THEN$(27:)=C$:RETURN:ELSE 5460
5470 PC=PC+1+L:CIM=PC:GOSUB 60:X=ADAT
5480 ON X/64+1 GOTO 5490,5500,10,5520
5490 RESTORE 8600:A=1+(X AND 5/8:B=1+(X AND 7):GOTO 5150
5500 V$(27:29)="BIT":GOTO 5530
5510 V$(27:29)="RES":GOTO 5530
5520 V$(27:29)="SET"
5530 B=(X AND 56)/8+30+((X AND)+1)*40:GOTO 5170
6000 W2$=CHR$(W2+48-7*(W2>9)):TURN
6010 W2=INT(W1/16):GOSUB 6000:$=W2$:W2=W1-16*W2:GOSUB 6000
6020 W1$=W1$&W2$:RETURN
6030 W1=INT(W/256):GOSUB 6010:=W1$:W1=W-256*W1:GOSUB 6010
6040 W$=W$&W1$:RETURN
6050 RESTORE 8700:KOD$=""
6060 FOR I=1 TO 14:READ X:KODKOD$&CHR$(X):NEXT I:RETURN
6070 CIM=CIM+(CIM>32767)*65536
6080 ADAT=USR(2+VARPTR(KOD$),CI):RETURN
6090 POKE VARPTR(KOD$)+4,PAGIRETURN
6100 Q2=ORD(Q2$)-48+7*(Q2$>"9:RETURN
6110 Q2$=Q1$(:1):GOSUB 6100:Q22:Q2$=Q1$(2:):GOSUB 6100
6120 Q1=16*Q1+Q2:RETURN
6130 Q1$=Q$(:2):GOSUB 6110:Q=Q1$=Q$(3:):GOSUB 6110
6140 Q=Q*256+Q1:RETURN
6150 SET INK 2:PRINT AT 6,5:";:SET INK TINTA:PRINT "ddress:"
6160 W=PC:GOSUB 6030:PRINT AT,13:W$:RETURN
6170 SET INK 2:PRINT AT 6,20:";:SET INK TINTA:PRINT "aging:"
6180 W1=PAGING:GOSUB 6010:PRI AT 6,27:W1$:RETURN
6190 SET INK 2:PRINT AT 7,5:";:SET INK TINTA:PRINT "yomtató:";
6200 PRINT AT 7,14:"";:ON NY TO 6210,6220
6210 PRINT "ki":RETURN
6220 PRINT "be":RETURN
6230 PRINT AT 4,10:">> ";
6240 ON MOD GOTO 6250,6260,62,6260,6260,6270,6280,6280,6320,6320
6250 PRINT "Disasm <<":RETURN
6260 RETURN
6270 PRINT " Word  <<":RETURN
6280 PRINT "Byte  <<":RETURN
6290 IF ADAT<32 OR ADAT>159 TN ADAT=46
6300 RETURN
6310 PRINT AT 7,19:"";:INPUT OMPT "Száma=":X:GOTO 3560
6320 PRINT "Text  <<":RETURN
6330 PRINT AT 23,2:"";:RETURN
6340 V$=STRING$(45,32):RETURN
8000 DATA B,C,D.E,H,L,(HL),A,C,DE,HL,SP,AF,(BC),(DE),(SP),(00),(0000)
8010 DATA 0000,0000,00,NZ,Z,NC,PO,PE,P,M,0,1,2,3,4,5,6,7,(HL),HL
8100 DATA 11,0,71,809,1,334,29,24,1,23,1,36,841,13,0
8110 DATA 33,533,22,371,1,5623,9,24,2,23,2,36,842,15,0
8120 DATA 66,19,71,810,1,3354,10,24,3,23,3,36,843,12,0
8130 DATA 65,19,22,411,1,6083,10,24,4,23,4,36,844,14,0
8140 DATA 65,782,71,811,71,4524,11,24,5,23,5,36,845,9,0
8150 DATA 65,783,22,451,71,7323,11,24,6,23,6,36,846,8,0
8160 DATA 65,784,71,812,71,3324,12,164,7,163,7,176,847,16,0
8170 DATA 65,785,22,491,71,7123,12,24,8,23,8,36,848,7,0
8200 DATA 22,48,22,88,22,1282,168,22,208,22,248,162,288,22,328
8210 DATA 27,48,27,88,27,1287,168,27,208,27,248,167,288,27,328
8220 DATA 20,1,20,2,20,3,20,20,5,20,6,160,7,20,8
8230 DATA 26,48,26,88,26,1286,168,26,208,26,248,166,288,26,328
8240 DATA 17,1,17,2,17,3,17,17,5,17,6,157,7,17,8
8250 DATA 21,1,21,2,21,3,21,21,5,21,6,161,7,21,8
8260 DATA 19,1,19,2,19,3,19,19,5,19,6,159,7,19,8
8270 DATA 18,1,18,2,18,3,18,18,5,18,6,158,7,18,8
8300 DATA 28,22,34,9,99,822,,20,95,822,10,9,57,848,32,17
8310 DATA 28,23,28,0,99,823,0,95,823,95,20,62,848,32,17
8320 DATA 28,24,34,10,99,8247,337,95,824,10,10,55,21,32,17
8330 DATA 28,25,6,0,99,825,3688,95,825,0,0,61,848,32,17
8340 DATA 28,26,34,11,99,8263,456,95,826,10,11,52,21,32,17
8350 DATA 28,27,29,38,99,8273,1570,95,827,0,0,56,21,32,17
8360 DATA 28,28,34,13,99,828,0,95,828,10,13,54,21,32,17
8370 DATA 28,29,1,452,99,825,0,95,829,0,0,53,21,32,17
8400 DATA LD,OUT,IN,DI,EI,EXCCF,CPL,DAA,PUSH,NOP,RLA,RLCA
8410 DATA RRA,RRCA,SCF,AND,COR,SUB,XOR,ADD,DEC,INC,CALL
8420 DATA SBC,ADC,RET,JP,JRJNZ,RST,EX,POP
8500 DATA NEG,68,RETN,69,IM00,"LD    I,A",71,RETI,77,"LD    R,A"
8510 DATA 79,IM1,86,"LD    A,",87,IM2,94,"LD    A,R",95,RRD
8520 DATA 103,RLD,111,LDI,16CPI,161,INI,162,OUTI
8530 DATA 163,LDD,168,CPD,16IND,170,OUTD,171,LDIR,176
8540 DATA CPIR,177,INIR,178IR,179,LDDR,184,CPIR,185
8550 DATA INDR,186,OTDR,187Not used",0
8600 DATA RLC,RRC,RL,RR,SLARA,SLL,SRL
8700 DATA 243,62,112,211,2,0,38,0,62,112,211,2,251,201
```

ISMERI ÖN A PRIMO-T?



Nyugodtan állíthatjuk, hogy a PRIMO-tulajdonosok gépükkel együtt sok gondot és bosszúságot is vásároltak. Márpedig az idei leértékeléskor nagyon sok gép talált gazdát. A felhasználói kézikönyv alapvető információkat sem közöl (pl. memóriatérkép, rendszerváltozók), a külön kapható hardver- és szoftverfüzettel a kezdő nehezen boldogul. Ezt az írásunkat a még nem vájtfülűeknek írjuk.

Talán már az is segítség, ha felsoroljuk, melyik korábbi BIT-LET számokban talál a PRIMO-s használható anyagot.

16-os szám (1985. jan.): A PRIMO vallatása, Karaktercsere
17: Futkározás (játék), Vallató hozzászólás
18: Hozzászólások
19: Hozzászólás
20: Vastagított betűk
22: Zene-bona, Információk (hang stb.)
25: Néhány rendszerváltozó
26: Kulcsszavak billentyűzetről
27: Kulcsszótáblázat, címek
28: (1986. jan.): A kommunikációs terület térképe
30: Monitorprogram
31: Rajz (Mandelbrot-halmazok)
33: Térbeli alakzatok rajza
36: Információk a gyártótól, Cirill betűk
38: Kulcsszavak egy billentyűnyomásra, A képernyő forgatása, eltolása (scroll)
40: (1987. jan.): Életjáték
42: Stopper óriás számjegyekkel
43: Átsorszámozó, Input rutin

A továbbiakban egy olyan középiskolai tanár adja közre PRIMO-s tapasztalatait, aki a HT–1080Z után ismerte meg ezt a gépet, majd a HT-vel, a Spectrummal összehasonlítva jött rá néhány fogásra, megoldásra.

1. A HT-s előzmények sokat segítettek – tanácsolom a kezdő PRIMÓ-soknak, hogy olvassanak HT-dokumentációt, BASIC-kézikönyvet: sok a hasonlóság. Kevés eltérés van a nyelvben (BEEP, CALL, CLOSE, CREATE, OPEN, PI, RESTORE n, RESUME n, RESUME NEXT, SAVE SCREEN, TEST), nagyon hasonlít a memóriatérkép, a rendszerváltozók címe, használata, a program, a változók és a tömbök tárolása, a stringek kezelése, a számábrázolás, a pontosság.

2. A tájékozódásban, különösen pedig a gépi kódú képernyőkezelés megértésében nyújthat segítséget a PRIMO RAM-jának vázlatos térképe, megjelölve néhány fontos címet, amelyen információt kaphatunk vagy beavatkozhatunk (1. ábra). Az adatok célszerű megváltoztatásával át lehet helyezni a BASIC-program kezdetét, a felhasználható terület végét (helyfoglalás gépi kódú programnak), a képernyőmemória kezdetét. A többi cím inkább informál, önkényes megváltoztatásuk zavart okozhat!

A beavatkozásnak természetesen szabályai vannak. Például a BASIC által elérhető terület felső határának áthelyezése után fontos egy CLEAR n utasítás, amely beállít több fontos rendszerváltozót. A programkezdet áthelyezése után pedig az új cím előtti byte-ra egy nullát kell tölteni, majd NEW parancsot adni. A helyfoglaláson kívül ilyen fogásokkal oldható meg több program egyesítése vagy egészen különleges memóriaáthelyezések (pl. Ötlet, 1985. aug. 29. – „Memóriabővítés").

3. A HT-használók ismerősként üdvözölhették a PRIMO nagyon praktikus DEF utasításait (Kézikönyv, 40–42.). Érdemes használni őket, sok billentyűnyomást lehet megtakarítani, esetenként gyorsabb programfutást, máskor nagyobb pontosságot érve el.

4. Ha PRIMO-felhasználó gépi kódban szeretne programozni, először meg kell ismernie a Z 80 mikroprocesszor utasításait. A gyakorlati kivitelezésnek aztán géptől függő problémái lesznek, például a gépi kódú programrész elhelyezése a memóriában. Ennek PRIMO-s lehetőségei:

a) A leginkább ajánlható módszer a BASIC-terület végén, a képernyőmemória előtt való elhelyezés. Ilyen megoldás látható az 1. programban. A legfontosabb előny az, hogy programból oldható meg, és a lefoglalt terület nagy lehet.

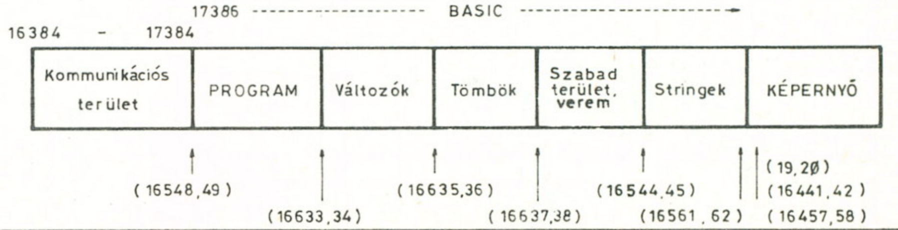
b) Helyet foglalhatunk a BASIC-program előtt is, de ez körülményesebb, mert a kezdőcím áthelyezését programon kívül, parancssal kell elvégezni.

c) Elterjedt módszer (a HT-s irodalomban talán túlságosan is), hogy a BASIC-program elején REM sor(ok)-ban foglalnak helyet. Előnye, hogy a programmal együtt kimenthető, hátránya viszont az, hogy a gépi kódú programban ilyenkor nem lehet nulla. A nullák kizárása viszont a program átszervezését igényli.

d) Jól el lehet helyezni rövidebb gépi rutinokat a kommunikációs terület egyes helyein is, megfelelő körültekintéssel. 36 összefüggő szabad byte található a 16476–16511-es címeken. A program futása közben szinte mindig szabad a kazettapuffer területe is (256 byte, 17129–17384). A billentyűpuffer (16872–17127) felső szakaszát is fel lehet használni, ha a programban csak rövid inputok vannak. Ugyancsak programot tölthetünk a lemezutasítások területére is (16722–16805), ennek meg az az ára, hogy közben nem írhatjuk be a CMD, FIELD stb. utasításokat.

A sakktáblarajzoló program gépi szubrutinját és a tábla 64 byte-os mintázatát a kazettapuffer területére töltjük. Ilyenkor nem kell területfoglalás, nem kell figyelembe venni az eltérő memóriaméreteket sem.

5. Az egyszerű BASIC-programok működését nem befolyásolja az, hogy milyen PRIMO-típuson hajtjuk végre. Vannak azonban olyan esetek, amelyekben nem közömbös a típus sem. Ha valaki több típust használ egyszerre (például az oktatásban) vagy publikálja programjait, bizonyos ese-



tekben külön intézkednie kell programjai hordozhatósága érdekében. Mikor fontos a típus ismerete, programbeli felismerése, lehetőleg automatikusan?

a) Ismernünk kell a gép memóriaméretét (–32, –48, –64), ha a képernyőmemória előtt akarunk helyet foglalni, vagy ha a művelet közvetlenül a képernyőmemóriára irányul (POKE vagy annak gépi megfelelője). Ennek programbeli felismerése és az ezzel való további vezérlés egyszerű: a ROM 19–20-as címén megtaláljuk a képernyőmemória kezdőcímét (a 19-esen 0 van, tehát elég a 20-as címet olvasni). A program előkészítő részében: MEM=PEEK(20). További összefüggések: a képernyő első byteja MEM*256; az utolsó MEM*256+6143; a BASIC által használható terület vége MEM*256–1; kétbyteos formában: 255, MEM–1. Mindkét szemléltető program felhasználja ezt, ME, illetve DI nevű változóval.

b) Ritkábban van szükség arra, hogy a program ismerje fel a gép A- vagy B típusát. Összehasonlítva a két típust, úgy látom, hogy az eltérések csak az eltérő karakterkészletből adódnak: a B típus négy ékezetes nagybetűvel többet ismer.

Mindkét változatban a 12791-es címen kezdődik a karakterek 8–8 byteon tárolt mintázata. Ha bármilyen okból fel akarjuk használni ezt a mintázatot (például nagyításra, újratervezésre), a négy karakternyi eltolódást észre kell vennie a programnak.

A kétféle ROM között számos ponton van eltérés, de a legjellemzőbb a 384-es tároló (A:49 – B:50). Ez íratja a bejelentkezéskor a 84.1, illetve 84.2 utolsó jelét. A 13832. és a 13841. címen pedig 30, illetve 26 jelzi az első karakterkódot (a 13800-on és a 13804-en 98, illetve 102 egészíti ezt ki 128-ra.) A 128-as kóddal kezdődő jelkészlet a B típuson néggyel kisebb: 147-ig tart csak.

A gyakorlatban célszerű így intézkedni: TIP=(PEEK(384)–49)*4. Így az A-t 0, a B-t 4 jelzi (négy karakter!). Az inverz szóköz kódja (ezt rajzokon jól lehet használni): 228–TIP. A 128-as karakter eredeti címe

```
10 REM ***********************************
20 REM *     A PRIMO-típus felismerése    *
30 REM ***********************************
40 REM >>> 1. A típus kiíratása          <<<
50 CLS: PRINT$ 2,0, CHR$(2) " PRIMO " CHR$
(PEEK(384)+16) "-" RIGHT$(STR$((PEEK(20)-1
04)/4+32),2) CHR$(18) " típusú gép vagyok"
: GOSUB 510
60 REM >>> 2. Helyfoglalás (97 byte)    <<<
70 RAMT=PEEK(20)*256-1: PRINT$ 5,6, "A BAS
IC-terület vége:" RA: RA=RA-97: PRINT$ 7,0
, "Lefoglalok 97 byte-ot, az új határ:" RA
: R%=RA/256: POKE 16561, RA-R%*256, R%: CL
EAR 50: GOSUB 510
80 REM >>> 3. Adatkeresés a ROM-ban     <<<
90 MEM=PEEK(20): TP=PEEK(384)-49: TIP=TP*4
100 REM >>> 4. Inverz szóköz képzése    <<<
110 PRINT$ 9,0, "Rajzolok egy inverz szókö
zt: " CHR$(228-TIP): GOSUB 510
120 REM >>> 5. Karakterminták olvasása <<<
130 CK=12791: PRINT$ 11,0, "Kiírom a ROM-b
ól egy karakter adatait.": INPUT "Melyik k
arakter adatait kéri"; C$: PRINT: PRINT
140 PE=CK+(ASC(C$)-PEEK(13832))*8: FOR C=0
 TO 7: PK=PEEK(PE+C): PRINT "   " PK;: FOR
 H=6 TO 2 STEP -1: HT=2↑H: KO=32: IF PK AN
D HT THEN KO=228-TIP
150 PRINT TAB(25-H*2) CHR$(KO);: NEXT: PRI
NT: NEXT: GOSUB 510
160 REM 6. Új karakter generálása        <<<
170 PRINT: PRINT: PRINT "Tervezek két jele
t (sakkbábu, görög pszí)"
180 CHAR=PEEK(16561)+PEEK(16562)*256+1: C%
=CH/256: POKE 16459, CH-C%*256, C%: CIM=CH
+(CH>32767)*65536: POKE CIM,144,56,16,56,1
6,56,124,125,0,68,84,84,56,16,16,17
190 PRINT "   " CHR$(128) "   " CHR$(129)
: GOSUB 510
200 REM >>> 7. A képernyő kezelése      <<<
210 CIM=CIM+16: X=MEM-1: Y=MEM+23: POKE CI
M,17,255,X,33,255,Y,1,32,0,237,184,17,255,
Y,33,223,Y,1,224,23,237,184,17,0,MEM,33,22
4,X,1,32,0,237,176,201
220 PRINT: PRINT "            Most scrollozo
k!": PRINT: PRINT "       A * billentyűre b
efejezem!"
230 E=CALL(CIM): BEEP 1,1: IF INKEY$<>"*"
THEN 230


240 REM >>> 8. Az adatok visszaállítása <<
250 POKE 16459, 7+TIP*8, 53: POKE 16561, 2
55, MEM-1: CLEAR 50: PRINT: PRINT "V é g e
": END
500 REM >>>>>>>>>>   Hangjelzés   <<<<<<<<<<
510 FOR C=0 TO 127: D=RND(10): BEEP D*20,2
0: NEXT: RETURN
```

```
10 REM ***********************************
20 REM *        S A K K T Á B L A         *
30 REM ***********************************
40 DEFINT A-J: A=255: B=256: DISP=PEEK(20)
: KEP=DI*B: KE=KE+(KE>32767)*B*B: CMIN=171
29: CPROG=CM+64: FRAM=16445
50 FOR I=0 TO 6 STEP 2: POKE CM+I*8,0,A,0,
A,0,A,0,A: POKE CM+(I+1)*8,A,0,A,0,A,0,A,0
: NEXT
60 POKE CP,17,233,66,33,0,DI,62,8,245,229,
14,8,26,213,17,30,0,6,24,119,35,119,35,119
,25,16,248,209,19,13,32,236,225,35,35,35,2
41,61,32,224
70 POKECP+40,33,224,DI-4,62,4,245,62,128,6
,24,17,0,3,25,17,32,0,25,119,16,252,241,61
,32,236,33,247,DI-1,62,4,245,62,1,6,24,17,
0,3,25,17,32,0,25,119,16,252,241,61,32,236
80 POKE CP+90,62,255,6,24,17,0,DI,33,224,D
I+23,18,119,19,35,16,250,201
100 REM >>>>>>> Rajzolás SET-tel <<<<<<<
110 S$="SET": GOSUB 510: FOR G=0 TO 144 ST
EP 48: FOR H=0 TO 144 STEP 48: FOR I=G TO
G+23: F=I+24: FOR J=H TO H+23: SET(J,I): S
ET(J+24,F): NEXT: NEXT: NEXT: NEXT
120 FOR I=24 TO 168 STEP 48: FOR J=0 TO 23
: E=I+J: F=E-24: SET(E,0): SET(F,191): SET
(0,E): SET(191,F): NEXT: NEXT: GOSUB 610
200 REM >>>>>>> Rajzolás POKE-kal <<<<<<
210 S$="POKE": GOSUB 510: FOR G=0 TO 18 ST
EP 6: FOR H=24 TO 168 STEP 48: FOR I=H TO
H+23: L=KE+I*32: M=KE+32*(I-24)+3: FOR J=G
 TO G+2: POKE L+J,A: POKE M+J,A: NEXT: NEX
T: NEXT: NEXT
220 FOR L=KE TO KE+4608 STEP 1536: FOR J=0
 TO 23: I=J*32: POKE L+I,128: POKE L+768+I
+23,1: NEXT: NEXT: FOR L=KE TO KE+23: POKE
 L,A: POKE L+6112,A: NEXT: GOSUB 610
300 REM >>>>>> Rajzolás gépi kóddal <<<<<.
310 S$="GEPI": GOSUB 510: C=CALL(CP): GOSU
B 610
320 PRINT$ 7,16, "V é g e !": END
500 REM (((((((( Az óra indítása ))))))))
510 CLS: PRINT$ 2,35,S$: BEEP 10,100: POKE
 FR,0,0,0: RETURN
600 REM ((((((( Az óra megállítása )))))))
610 H=PEEK(FR): J=PEEK(FR+1): BEEP 100,100
: X=H+J*B: EP=X/6000: EM=X-EP*6000: Y=EM/1
00: PRINT$ 5,33, EP "perc": PRINT$ 7,33, U
SING "##.## mp"; Y
620 IF INKEY$="T" THEN CLS ELSE 620
630 IF INKEY$="R" THEN RETURN ELSE 630
```

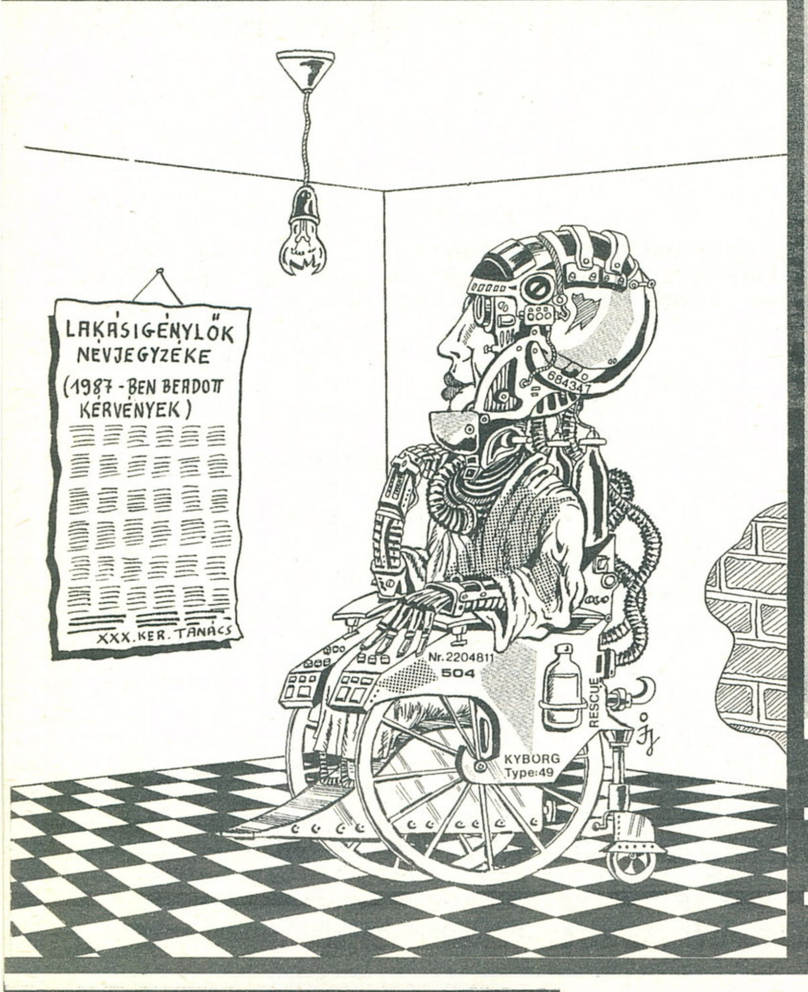
(amelyet megváltoztatunk a 16459–60-as címen, ha mi tervezünk saját jeleket, de amelyet később vissza kell állítanunk): 7+TIP∗8, 53.

SEGÉDPROGRAMOK

A leírtak illusztrálására két programot ajánlok. Az első sorra bemutatja az eltérő típusú PRIMO-k felismerésének különböző fogásait, gyakorlati alkalmazásait. Ezekre a megértés után más ötletek építhetőek (betűnagyítás, játékok képernyőmozgatása stb.). A második programot ugyancsak gondolatébresztőnek szánom, miközben bemutatom egy feladat megoldásának három különböző szintjét, minőségét. A feladat látványos: egész képernyőt betöltő sakktábla rajza. A program háromféle eszközzel oldja meg: SET, POKE, gépi kódú szubrutin. Közben működteti és olvassa a belső órát is, majd kiírja, mennyi idő kellett a rajzhoz. Meggyőző az eltérés! Egy-egy rajz után SHIFT+T-vel lehet letörölni a képet, az újabb rajzolást pedig SHIFT+R-rel lehet indítani.

A SET-tel való rajzolás programozása egyszerű, a munka azonban nagyon lassú. A képernyőmemóriába való közvetlen beavatkozás (POKE) egyszerre 8 pontot rajzol, kb. ötször gyorsabb. A gépi kódú munka sebessége még ennek is csaknem 2000-szerese.

Fekete György
7300 Komló, Bocskai u 30.



SZOFTVER ÖTLETEK



HANGADÓ PLUS/4 ÉS C 16

A program rövid ugyan, de nagyon jó hanghatásokat és dallamokat lehet vele elérni. Fel lehet használni programokhoz is, kísérőzenékhez.

A program átírja a megszakításvektort, így BASIC vezérlést nem igényel. A zene addig szól, ameddig a megszakításvektort vissza nem írjuk.

A programot **$2000**-re helyeztem, de természetesen máshova is lehet rakni. Ekkor a megszakításvektort értelemszerűen át kell írni.

A programot monitorban írhatjuk be "A" parancs segítségével.

A program indítása monitorból: "G 2000", **BASIC-ből** SYS2∗4096. A $2000-től $200D-ig tartó rész végzi a megszakításvektor átírását. A tulajdonképpeni program $2000-től $2010-ig tart. Ez csökkenti a $FF11 cím tartalmát, és elmegy az eredeti megszakításkezelőre.

A $FF11 cím a hangvezérlő regiszter. E byte bitjeinek jelentése:

- B7. 0 esetén a hang időzítés nélkül, folyamatos
- B6. 1 esetén a 2. hanggenerátor zaj engedélyezett
- B5. 1 esetén a 2. hanggenerátor négyszögjel engedélyezett
- B4. 1 esetén az 1. hanggenerátor engedélyezett
- B3–B0 hangerő értéke (csak 8-ig hatásos)

A megszólaló hangok magasságát SOUND utasítással állíthatjuk be (pl. SOUND1,200,1:SOUND3,900,1)

BASIC-vezérlés:

Kikapcsolás:

```
POKE DEC("2002"),DEC("0E"):POKE DEC("2007"),
DEC("CE"):SYS2∗4096:REM OFF
```

Bekapcsolás:

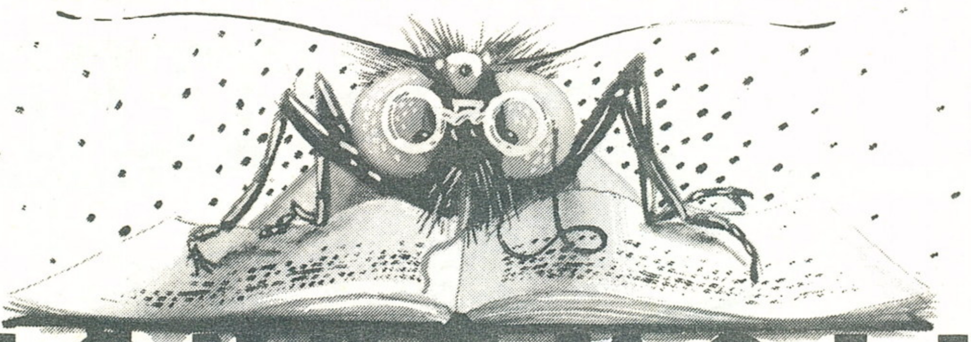
```
POKE DEC("2002"),DEC("0D"):POKE DEC("2007"),
DEC("20"):SYS 2∗4096:REM ON
```

A programot érdemes kipróbálni, mert lényegesen jobb hanghatás érhető el vele, mint a BASIC SOUND utasítással. Eredményes kísérletezést és jó szórakozást kíván:

Hábeller Zsolt, 9600 Sárvár, Bartók Béla út 5.

```
. 2000  78        SEI
. 2001  A9 0D     LDA#$0D
. 2003  8D 14 03  STA0314
. 2006  A9 20     LDA#$20
. 2008  8D 15 03  STA0315
. 200B  58        CLI
. 200C  60        RTS
. 200D  CE 11 FF  DECFF11
. 2010  4C 0E CE  JMPCE0E
```

A szerkesztő azért van, hogy a lap olyan legyen, amilyenek az olvasói!



KÖNYVMOLY

Werner Rügemer: A Szilícium-völgy – Kossuth Könyvkiadó, 216 o., 50 Ft.
(Riportkötet a mikroelektronikai ipar központjának működéséről, életéről.)

Vadnai Szabolcs: Commodore 16 Programozói zsebkönyv – Novotrade, 78 o., 198 Ft.
(A C–16 működésének és kezelésének bemutatása, a C 64-es programozói zsebkönyvhöz hasonló, kézikönyvszerű formában.)

Steigers: A robotok és a Commodore 64 – DATA BECKER – Novotrade, 128 o., 249 Ft.
(A könyv hasznos tanácsokat nyújt ahhoz, hogy C–64-es gépünket hogyan használhatjuk kis robotok vezérlésére, irányítására.)

Markus Weber: IBM PC 3D-grafika – IWT – Novotrade, 184 o., 380 Ft.
(Az erősen matematikai szemléletű kötet a térbeli grafikai alkalmazás alapjaiba vezet be.)

A hazai mikroszámítógépes szoftverpiac és az IBM szoftver-kínálat – Száminform Gmk, 84 o.

Az IBM-felhasználók bizonyára haszonnal forgatják a kötetet. A címlapon szerepel, hogy ez a II. szerkesztés, kiadási dátumként pedig 1987. III. negyedév – így minden arra utal, hogy végre egy naprakész kiadványt tarthatunk a kezünkben. Ez a kötet egy dologban biztos eltér minden eddig általunk értékelt kötettől. Üzletben ugyanis nem vásárolható. Kis példányszámban jelent meg, és csak a kiadónál rendelhető. Ára is meglehetősen borsos, (mintegy 5500 Ft), mivel megjelentetői úgy gondolták, hogy akinek telik egy IBM-gépre, az nem fog garasoskodni egy szoftverkatalógus megvételénél sem. Igaz ami igaz, nem mindenkinek az elveivel egyeztethető össze az a filozófia, hogy az információt annak felhasználója kell, hogy megfizesse. Mert az alapgondolat ugyan vitathatatlan, de bosszantó, hogy a felhasználó kétszer fizet – egyszer az információkért, másodszor pedig, ha a kívánt programot kiválasztotta és megrendelte. A szoftver eladója viszont csak kasszíroz, hiszen nem járult hozzá egy fityinggel sem annak a kiadványnak az elkészültéhez, amely pedig ha úgy vesszük reklámot jelent neki. A kiadó gmk-képviselőivel hosszú vitát folytattunk erről, s abban igazat kell nekik adnunk, hogy csak így biztosítható, hogy az információkat összefoglaló cég ne legyen egyik programkészítőnek sem lekötelezve.
E gépeket egyébként is általában nem magánszemélyek, hanem vállalatok, intézmények vásárolják, ahol egy jó katalógus ára bőven megtérülhet, ha okosan használják fel – és ha a katalógus valóban a kívánt információkat tartalmazza. A kérdés most már csak annyi, hogy van-e igazán értelme ennek a befektetésnek, azaz nyújt-e ez a kötet annyit, hogy egy-egy cégnek megérje megrendelni, megvenni. (A kispénzű felhasználók számára azért hadd jegyezzünk meg annyit, hogy a kötet az Országos Műszaki Könyvtárban hozzáférhető.)
Honnan szerezhet még valaki e köteten kívül információt arról, hogy milyen szoftverek kaphatók ma Magyarországon? A KSH-nak volt egy kiadványa még 1982-ben, „Hazai szoftver-kínálat" címmel. Ezt igyekeztek évenként megjelenő kötetté fejleszteni – de a próbálkozás elhalt. A stafétát az LSI ATSZ vette át, amely 1984-ben egy, 1985-ben két, 1986-ban pedig újra egy katalógust adott ki. Az 1984-es katalógus általános, itt a kiadó a lehető legátfogóbb bemutatásra törekedett. A programok ismertetésére téma szerint került sor, és bekerült ABC 80 programtól a TPA-n futtathatóig minden. Ez nem a szerkesztő vagy a kiadó hibája. Pusztán arról van szó, hogy 1984-ben az összes, Magyarországon forgalmazott szoftver ismertetése – bármilyen gépre – belefért egyetlen vaskos kötetbe.
A következő LSI-kötet 1985-ben még ugyanezt a gondolkodásmódot követte; de ebben az évben megjelent az a kiadvány is, amely már csak az egészségüggyel, az irodagépesítéssel és az adatfeldolgozással foglalkozott. Vagyis szűkebb területet ölelt fel, de így képes volt a részletesebb bemutatásra is. Az 1986-os kötet pedig már géptípus szerint specilizálódott. Erre utal az alcím is: IBM és Commodore család.
Mi újat nyújthat e kötetekhez képest a Száminform kiadványa? Sajnos, nagyon sokat – és nagyon keveset. Sokat, mert ez a könyv, és a más kiadók által megjelentetett korábbiak között alig van átfedés. Vagyis egy-egy program vagy az egyik könyvben szerepel, vagy a másikban, de csak ritkán mindkettőben. Ez, persze betudható annak is, hogy a Száminform igyekezett a legfrissebb információkat közölni. Így azután lehet, hogy keveset is nyújt: nem hisszük, hogy minden egy-két éves szoftver annyira elavult volna, hogy említést sem érdemel.

A Száminform kötete mégis az az ismertetés, ami igyekszik a legszélesebb körű lenni. Emiatt több olyan hibába is beleesik, amiről e rovatban már szóltunk. Így a kötet java része függelék, a programok felsorolása. Ez, persze természetes, ha egy könyvnek az a célja, hogy a hazai szoftverkínálatot mutassa be – de akkor miért kell ezt a függelékben megtenni? Furcsa szerkezetű könyv az, amelynek közel háromnegyed részét a függelék teszi ki.

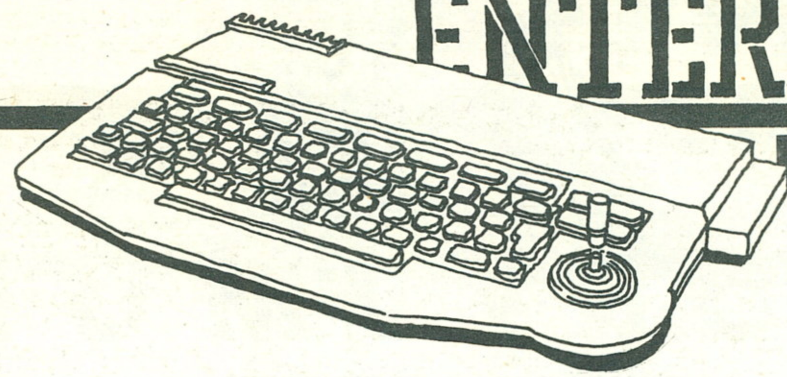
A másik hiba nem csupán szerkesztési, de érinti a tartalmat is: jó, hogy megismerjük egy-egy program nevét, de a névből gyakran nem derül ki az, hogy mire is jó a program. Emiatt jobb az LSI kötetek szerkezete: itt minden programra rászánnak egy-egy oldalt, és ismertetik annak célját, működését is. Egy többszáz programot felvonultató kiadványtól persze nem várható el ez; de kérdés, hogy ha a kötetet több részben, témák szerint csoportosítva, és részletesebb ismertetéssel együtt adták volna ki, akkor nem lenne-e hasznosabb és sikeresebb?

Idáig azonban még mindig csak a „Függelék"-ről szóltunk – annak ellenére, hogy a kötet bevezető részének is sok erénye van. Ez egy piacszemléletű fejtegetés, ami kitér arra, hogy az elmúlt években hogyan alakult az IBM-kompatíbilis gépek kereslete és kínálata, az árak és a piaci tájékozódás. Mindezen vizsgálatok eredményeit a szerzők táblázatokkal és diagrammokkal támasztják alá. Efféle adatokat a korábban említett művek nem tartalmaznak, így joggal állíthatjuk, hogy ez az egyetlen könyv, amiből ilyen – korántsem haszontalan – információk birtokába juthatunk.

Tallér József

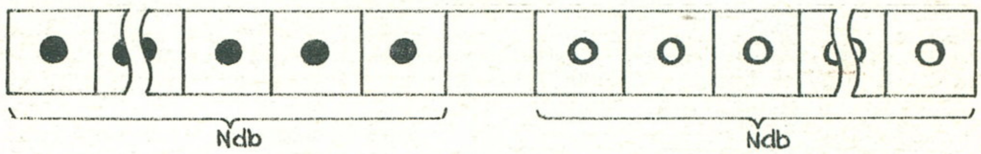
CENTRUM

ENTERPRISE® nyerő



Mostani feladatunk nem minden előzmény nélküli. Több mint három éve (1984 áprilisában) 5 gép nyerő pályázatunkban közöltünk egy feladatot, amely tulajdonképpen ennek a mostaninak a rokona, előzménye.

Tegyük fel, hogy van egy játékterünk, amely mondjuk tizenegy egymás melletti négyzetből áll. A mező egyik oldalán öt sötét színű korong áll, a másik oldalon szemben vele öt világos színű. A középső mező tehát üres. Mindezt mutatja ábránk is. A feladat az, hogy minél kevesebb lépéssel cseréljük meg a világos és sötét korongokat. Természetesen a táblát nem szabad megmozdítani (például találékonyságra vallana a feladat egyszerű megoldása a tábla elforgatásával). A következők szerint léphetünk:



- **Egy sötét koronggal léphetünk egyet jobbra, ha az a mező üres.**
- **Egy világos koronggal léphetünk egyet balra, ha az a mező üres.**
- **Egy sötét koronggal átugorhatjuk a jobb oldalán álló világos korongot, (csak egyet!) ha ennek szomszédján üres mező van.**
- **Egy világos koronggal átugorhatjuk a bal oldalán álló sötét korongot, (csak egyet!) ha ennek szomszédján üres mező van.**

Röviden ennyi a játék. Eddig 11 mezőből álló pályáról beszéltünk. De természetesen bármilyen páratlan számú mezővel megadható a pálya mérete, s ennek megfelelően módosul a korongok száma. **S ennek megfelelően maga a feladat:**

Írjanak egy minél rövidebb és szebb BASIC programot, amely megkérdezi N értékét. N alapján 2N+1 méretű játéktérre és ennek megfelelően N világos és N sötét korongra elvégzi a következőket:

Megoldja a játékot, azaz kiírja egymás után a szükséges lépéseket, a képernyőn pedig mutatja a pillanatnyi állapotot. Természetesen a képernyő mérete határt szab a megmutathatóságnak. Elegendő ha programjuk $N \leq 15$-ig tudja a megjelenítést, s természetesen a karaktergrafikát ajánljuk figyelmükbe a program terjedelme miatt is.

A megírt programokat a szükséges magyarázatokkal, leírásokkal együtt ezúttal is papíron kérjük beküldeni!

ENTERPRISE® nyerő

Kérjük levágni és a levélre felragasztani! Beküldési határidő: 1987. november 15.

BIT-LET

Szerkesztőségünk címe: ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál. Tördelő: Domokos Imre. Hírrovat: Csányi György. Posta: Halász Péter.